

ГЛАВА 2. МЕТОДИЧЕСКИЙ АНАЛИЗ РЕЗУЛЬТАТОВ ЕГЭ ПО ИНФОРМАТИКЕ

РАЗДЕЛ 1. ХАРАКТЕРИСТИКА УЧАСТНИКОВ ЕГЭ ПО ИНФОРМАТИКЕ

1.1. Количество участников ЕГЭ по учебному предмету (за 3 года)

Таблица 2-1

2024 г.		2025 г.		2026 г.	
чел.	% от общего числа участников	чел.	% от общего числа участников	чел.	% от общего числа участников
1432	14,84	1560	15,42	1605	15,15

1.2. Процентное соотношение юношей и девушек, участвующих в ЕГЭ (за 3 года)

Таблица 2-2

Пол	2024 г.		2025 г.		2026 г.	
	чел.	% от общего числа участников	чел.	% от общего числа участников	чел.	% от общего числа участников
Женский	328	22,91	357	22,88	363	22,62
Мужской	1104	77,09	1203	77,12	1242	77,38

1.3. Количество участников экзамена в регионе по категориям (за 3 года)

Таблица 2-3

Категория участника	2024 г.		2025 г.		2026 г.	
	чел.	% от общего числа участников	чел.	% от общего числа участников	чел.	% от общего числа участников
ВТГ, обучающихся по программам СОО	1392	97,21	1518	97,31	1545	96,26
ВТГ, обучающихся по программам СПО	4	0,28	10	0,64	15	0,93
ВПЛ	36	2,51	32	2,05	45	2,80

1.4.Количество участников экзамена в регионе по типам ОО

Таблица 2-4

№ п/п	Категория участика	2024 г.		2025 г.		2026 г.	
		чел.	% от общего числа участников	чел.	% от общего числа участников	чел.	% от общего числа участников
1.	Средняя общеобразовательная шко- ла	724	50,56	861	55,19	866	53,96
2.	Средняя общеобразовательная шко- ла с углубленным изучением отдель- ных предметов	103	7,19	92	5,90	114	7,10
3.	Гимназия	272	18,99	242	15,51	256	15,95
4.	Лицей	246	17,18	274	17,56	268	16,70
5.	Средняя общеобразовательная шко- ла-интернат	0	0,00	2	0,13	0	0,00
6.	Лицей-интернат	40	2,79	42	2,69	33	2,06
7.	Кадетская школа-интернат	0	0,00	0	0,00	1	0,06
8.	Общеобразовательная школа-интер- нат с первоначальной летной подго- товкой	2	0,14	2	0,13	4	0,25
9.	Открытая (сменная) общеобразова- тельная школа	4	0,28	3	0,19	2	0,12
10.	Техникум	1	0,07	1	0,06	1	0,06
11.	Иное	40	2,79	41	2,63	60	3,74

1.5.Количество участников ЕГЭ по учебному предмету по АТЕ региона

Таблица 2-5

№ п/п	Наименование АТЕ	Количество участников ЕГЭ по учебному предмету	% от общего числа участни- ков в регионе
1	Алейский район	2	0,12
2	Алтайский район	20	1,25
3	Баевский район	4	0,25
4	Бийский район	13	0,81

№ п/п	Наименование АТЕ	Количество участников ЕГЭ по учебному предмету	% от общего числа участни- ков в регионе
5	Благовещенский район	18	1,12
6	Бурлинский район	3	0,19
7	Быстроистокский район	2	0,12
8	Волчихинский район	5	0,31
9	Егорьевский район	2	0,12
10	Ельцовский район	4	0,25
11	Завьяловский район	5	0,31
12	Залесовский муниципальный округ	9	0,56
13	Змеиногорский район	11	0,69
14	Заринский район	2	0,12
15	Зональный район	8	0,50
16	Калманский район	4	0,25
17	Каменский район	22	1,37
18	Ключевский район	5	0,31
19	Косихинский район	2	0,12
20	Красногорский район	3	0,19
21	Краснощековский район	6	0,37
22	Крутихинский район	1	0,06
23	Кулундинский район	6	0,37
24	Курынский район	3	0,19
25	Кытмановский район	4	0,25
26	Локтевский район	9	0,56
27	Мамонтовский район	15	0,93
28	Михайловский район	11	0,69
29	Немецкий национальный район	1	0,06
30	Новичихинский район	2	0,12
31	Павловский район	12	0,75
32	Панкрушихинский район	1	0,06
33	Первомайский район	21	1,31
34	Петропавловский район	3	0,19
35	Поспелихинский район	17	1,06

№ п/п	Наименование АТЕ	Количество участников ЕГЭ по учебному предмету	% от общего числа участни- ков в регионе
36	Ребрихинский район	11	0,69
37	Родинский район	7	0,44
38	Романовский район	5	0,31
39	Рубцовский район	10	0,62
40	ЗАТО Сибирский	3	0,19
41	Смоленский район	15	0,93
42	Советский район	5	0,31
43	Солонешенский район	4	0,25
44	Солтонский район	1	0,06
45	Суетский район	1	0,06
46	Табунский район	2	0,12
47	Тальменский район	14	0,87
48	Тогульский район	6	0,37
49	Топчихинский район	2	0,12
50	Третьяковский район	3	0,19
51	Троицкий район	2	0,12
52	Тюменцевский район	4	0,25
53	Угловский район	3	0,19
54	Усть-Калманский район	1	0,06
55	Усть-Пристанский район	3	0,19
56	Хабарский район	2	0,12
57	Целинный район	3	0,19
58	Чарышский район	3	0,19
59	Шипуновский район	8	0,50
60	Шелаболихинский район	2	0,12
61	г. Алейск	15	0,93
62	г. Барнаул	736	45,86
63	г. Белокуриха	8	0,50
64	г. Бийск	173	10,78
65	г. Заринск	24	1,50
66	г. Новоалтайск	55	3,43

№ п/п	Наименование АТЕ	Количество участников ЕГЭ по учебному предмету	% от общего числа участни- ков в регионе
67	г. Рубцовск	96	5,98
68	г. Славгород	33	2,06
69	г. Яровое	11	0,69
70	Краевые образовательные организации	70	4,36
71	Негосударственные образовательные организации	8	0,50

1.6. Прочие характеристики участников экзаменационной кампании (при наличии) нет

1.7. ВЫВОДЫ о характере изменения количества участников ЕГЭ по учебному предмету

На основе приведенных в разделе данных отмечается стабильный рост абсолютного числа участников ЕГЭ по информатике в 2026 году, относительно прошлых лет. Доля участников ЕГЭ по информатике относительно общего числа выпускников образовательных организаций сохраняется на уровне 2025 года.

Это можно объяснить рядом причин:

четкой и понятной системой проведения компьютерного ЕГЭ по информатике, удачно апробированной в 2021 году с предоставлением потенциальным участникам тестирования возможности тренировать экзамен в эмуляторе станции КЕГЭ;

сохранение популярности сферы ИТ для выбора профессий, а также государственной поддержкой ИТ-отрасли;

трендом на развитие цифрового сектора экономики в стране;

активным развитием в регионе направлений работы со школьниками в системе дополнительного образования в области ИТ, ИИ, робототехники, управления беспилотными летательными аппаратами и пр. (в точках роста, на кружках, в технопарках – ИТ-куб, Таланты22, Педагогический технопарк АлтГПУ «Кванториум», центр «Наследники Ползунова» АлтГТУ при поддержке Благотворительного Фонда Андрея Мельниченко).

Доли экзаменуемых женского и мужского пола остались теми же, что и в 2025 году.

Как и в предыдущие годы, основную часть участников ЕГЭ по информатике составили выпускники текущего года, обучавшиеся по программам среднего общего образования (96,26%). Преимущественно это выпускники СОШ, СОШ с углубленным изучением предмета, лицеев и гимназий.

Сохраняется доминирование числа городских обучающихся над сельскими, что объясняется рядом причин – выбор информатики и не обязателен для сдачи ЕГЭ, в вузах введена практика выбора альтернативных вступительных

экзаменов, на селе не хватает специалистов данного профиля, профильное обучение в сельских школах с малым количеством обучающихся вести невозможно, а ЕГЭ по информатике требует углубленной подготовки по предмету, предусмотренной для программ обучения по соответствующему профилю. Всего городских участников экзамена по информатике 71,73%.

Как и в 2025 среди участников ЕГЭ по информатике 2026 года только на г. Барнаул и г. Бийск приходится больше половины экзаменуемых – 56,64%. Таким образом, можно констатировать сохранение проблем, связанных с отсутствием предметников и низким качеством подготовки по информатике на селе, и, наличием возможностей для профильной подготовки по информатике в городских ОО, в большей степени на это влияет наличие подготовленных кадров в городах и открытие IT-кубов, технопарков, кванториумов и пр.

РАЗДЕЛ 2. ОСНОВНЫЕ РЕЗУЛЬТАТЫ ЕГЭ ПО ПРЕДМЕТУ

2.1. Диаграмма распределения тестовых баллов участников ЕГЭ по предмету в 2026 г. (количество участников, получивших тот или иной тестовый балл)

1) Распределение тестовых баллов

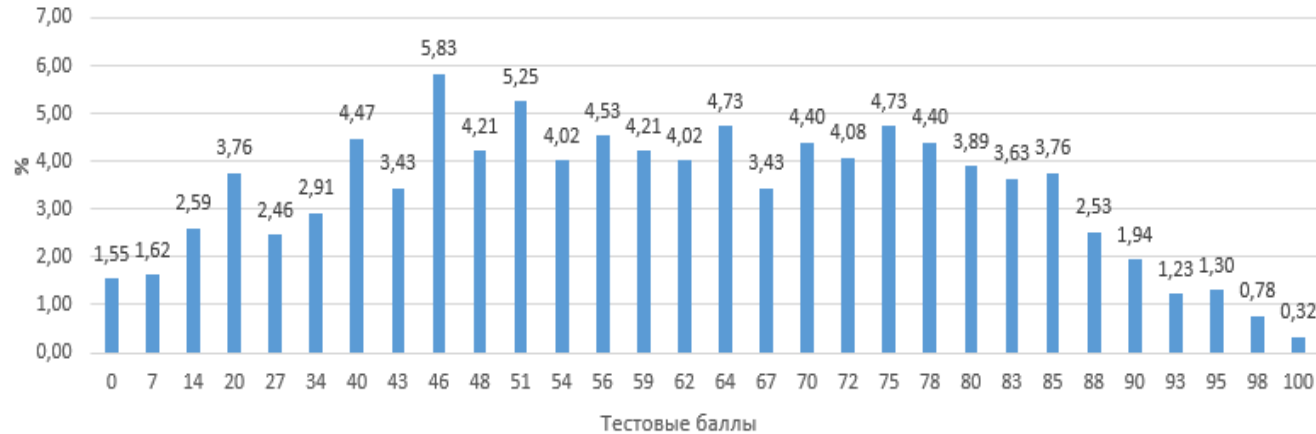


Рисунок 2-1. Распределение тестовых баллов

2) Сгруппированное распределение тестовых баллов

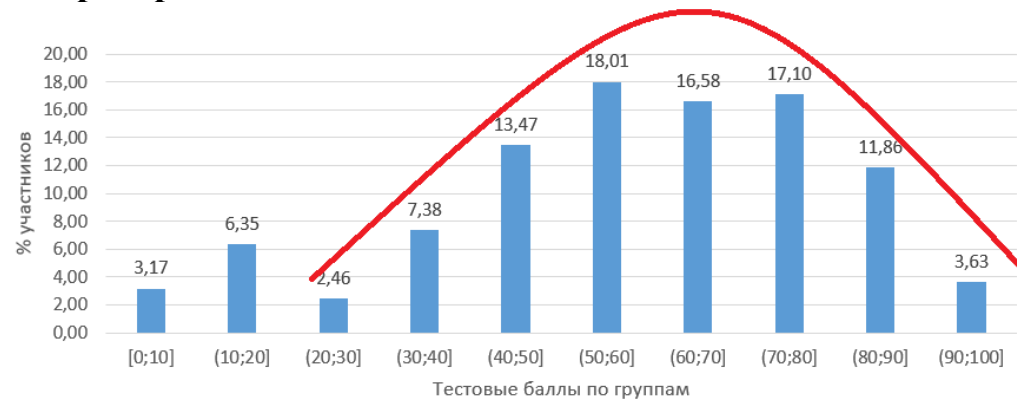


Рисунок 2-2. Сгруппированное распределение тестовых баллов

2.2.Динамика результатов ЕГЭ по предмету за последние 3 года

Таблица 2-6

№ п/п	Участников, набравших балл	Год проведения ГИА		
		2024 г.	2025 г.	2026 г.
1.	ниже минимального балла, %	16,83	17,25	15,90
2.	от минимального балла до 60 баллов, %	43,09	42,14	36,22
3.	от 61 до 80 баллов, %	30,31	28,29	32,86
4.	от 81 до 100 баллов, %	9,78	12,32	15,02
5.	Средний тестовый балл	54,21	55,37	57,63

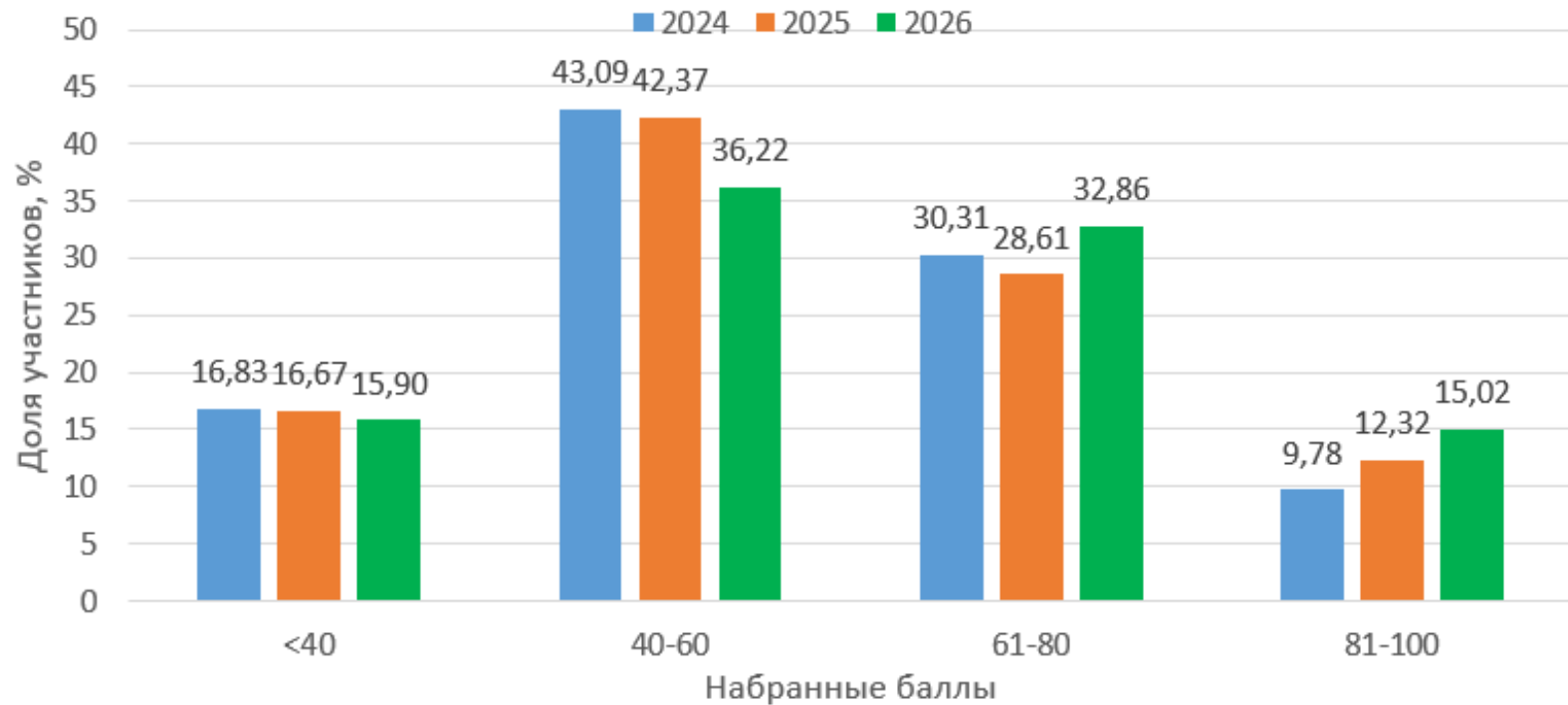


Рисунок 2-3. Сгруппированное распределение тестовых баллов в динамике за 3 года

2.3. Результаты ЕГЭ по учебному предмету по группам участников экзамена с различным уровнем

ПОДГОТОВКИ**2.3.1.** в разрезе категорий участников ЕГЭ*Таблица 2-7*

№ п/п	Категории участников	Доля участников, у которых полученный тестовый балл			
		ниже минимального	от минимального балла до 60 баллов	от 61 до 80 баллов	от 81 до 100 баллов
1.	ВТГ, обучающиеся по программам СОО	1544	14,9	35,95	33,68
2.	ВТГ, обучающиеся по программам СПО	15	33,33	46,67	13,33
3.	ВПЛ	45	44,44	42,22	11,11

2.3.2. в разрезе типа ОО*Таблица 2-8*

№ п/п	Тип ОО	Количество участников, чел.	Доля участников, получивших тестовый балл			
			ниже минимального	от минимального до 60 баллов	от 61 до 80 баллов	от 81 до 100 баллов
1.	СОШ	865	20,12	40,23	30,29	9,36
2.	СОШ с УИОП	114	12,28	31,58	40,35	15,79
3.	Гимназии, лицеи	524	6,87	31,11	37,02	25
4.	Интернаты	34	8,82	11,76	52,94	26,47
5.	Вечерние и открытые (сменные) ОШ	2	100	0	0	0
6.	Другие	5	20	80	0	0

2.3.3. в разрезе уровня изучения предмета*Таблица 2-9*

№ п/п	Уровень изучения предмета	Количество участников, чел.	Доля участников, получивших тестовый балл			
			ниже минимального	от минимального до 60 баллов	от 61 до 80 баллов	от 81 до 100 баллов
1.	базовый	1087	17,85	39,1	31,09	11,96
2.	углубленный	457	7,88	28,45	39,82	23,85

2.3.4. в сравнении по АТЕ

Таблица 2-10

№ п/п	Наименование АТЕ	Количество участников, чел.	Доля участников, получивших тестовый балл			
			ниже минималь- ного	от минималь- ного до 60 бал- лов	от 61 до 80 бал- лов	от 81 до 100 бал- лов
1.	Алейский район	2	0	50	0	50
2.	Алтайский район	20	30	40	30	0
3.	Баевский район	4	0	75	25	0
4.	Бийский район	12	25	50	25	0
5.	Благовещенский район	18	50	22,22	27,78	0
6.	Бурлинский район	3	33,33	33,33	0	33,33
7.	Быстроистокский район	2	0	0	50	50
8.	Волчихинский район	5	0	100	0	0
9.	Егорьевский район	2	0	0	100	0
10.	Ельцовский район	4	25	75	0	0
11.	Завьяловский район	5	20	40	40	0
12.	Залесовский муниципальный округ	9	22,22	33,33	33,33	11,11
13.	Змеиногорский район	11	18,18	63,64	18,18	0
14.	Заринский район	2	0	100	0	0
15.	Зональный район	8	25	50	12,5	12,5
16.	Калманский район	4	0	25	25	50
17.	Каменский район	22	22,73	31,82	36,36	9,09
18.	Ключевский район	5	0	80	0	20
19.	Косихинский район	2	0	100	0	0
20.	Красногорский район	3	33,33	0	66,67	0
21.	Краснощековский район	6	0	33,33	50	16,67
22.	Крутихинский район	1	0	100	0	0
23.	Кулундинский район	6	33,33	33,33	33,33	0
24.	Курьинский район	3	66,67	33,33	0	0
25.	Кытмановский район	4	0	0	50	50
26.	Локтевский район	9	22,22	22,22	55,56	0
27.	Мамонтовский район	15	6,67	66,67	20	6,67
28.	Михайловский район	11	0	27,27	63,64	9,09

№ п/п	Наименование АТЕ	Количество участников, чел.	Доля участников, получивших тестовый балл			
			ниже минималь- ного	от минималь- ного до 60 бал- лов	от 61 до 80 бал- лов	от 81 до 100 бал- лов
29.	Немецкий национальный район	1	100	0	0	0
30.	Новичихинский район	2	0	100	0	0
31.	Павловский район	12	8,33	66,67	8,33	16,67
32.	Панкрушихинский район	1	0	0	0	100
33.	Первомайский район	21	14,29	66,67	19,05	0
34.	Петропавловский район	3	66,67	0	33,33	0
35.	Поспелихинский район	17	5,88	29,41	52,94	11,76
36.	Ребрихинский район	10	60	30	10	0
37.	Родинский район	7	0	57,14	42,86	0
38.	Романовский район	5	60	20	0	20
39.	Рубцовский район	8	37,5	62,5	0	0
40.	ЗАТО Сибирский	3	33,33	0	0	66,67
41.	Смоленский район	14	21,43	14,29	42,86	21,43
42.	Советский район	5	20	20	60	0
43.	Солонешенский район	4	0	50	25	25
44.	Солтонский район	1	0	0	100	0
45.	Суетский район	1	0	100	0	0
46.	Табунский район	2	0	50	50	0
47.	Тальменский район	14	21,43	57,14	21,43	0
48.	Тогульский район	6	33,33	33,33	33,33	0
49.	Топчихинский район	2	0	0	100	0
50.	Третьяковский район	3	33,33	0	33,33	33,33
51.	Троицкий район	2	0	100	0	0
52.	Тюменцевский район	4	25	75	0	0
53.	Угловский район	3	0	66,67	0	33,33
54.	Усть-Калманский район	1	0	100	0	0
55.	Усть-Пристанский район	3	0	66,67	33,33	0
56.	Хабарский район	2	50	0	50	0
57.	Целинный район	3	33,33	33,33	33,33	0

№ п/п	Наименование АТЕ	Количество участников, чел.	Доля участников, получивших тестовый балл			
			ниже минималь- ного	от минималь- ного до 60 бал- лов	от 61 до 80 бал- лов	от 81 до 100 бал- лов
58.	Чарышский район	3	33,33	33,33	33,33	0
59.	Шипуновский район	8	12,5	25	62,5	0
60.	Шелаболихинский район	2	50	50	0	0
61.	г. Алейск	15	20	13,33	46,67	20
62.	г. Барнаул	695	10,5	31,8	37,84	19,86
63.	г. Белокуриха	8	50	12,5	37,5	0
64.	г. Бийск	162	17,9	43,21	28,4	10,49
65.	г. Заринск	23	17,39	21,74	39,13	21,74
66.	г. Новоалтайск	54	12,96	48,15	29,63	9,26
67.	г. Рубцовск	95	18,95	45,26	28,42	7,37
68.	г. Славгород	32	18,75	34,38	31,25	15,63
69.	г. Яровое	11	9,09	27,27	45,45	18,18
70.	Краевые образовательные организа- ции	70	5,71	17,14	38,57	38,57
71.	Негосударственные образовательные организации	8	50	37,5	0	12,5

2.4. Выделение перечня ОО, продемонстрировавших наиболее высокие и низкие результаты ЕГЭ по предмету

2.4.1. Перечень ОО, продемонстрировавших наиболее высокие результаты ЕГЭ по предмету

Таблица 2-11

№ п/п	Наименование ОО	Количество ВТГ, чел.	Доля ВТГ, получивших тестовый балл			
			от 81 до 100 бал- лов	от 61 до 80 бал- лов	от минималь- ного балла до 60 баллов	ниже минималь- ного
1	МАОУ "СОШ №132" им. Н.М. Мала- хова (г. Барнаул)	11	72,73	27,27	0,00	0,00
2	МБОУ "Лицей №124" (г. Барнаул)	59	61,02	38,98	0,00	0,00
3	КГБОУ "Алтайский краевой педаго- гический лицей-интернат» (АКПЛ)	31	58,06	29,03	12,90	0,00

	(краевые образовательные организации)					
4	МБОУ "СОШ №15" (г. Славгород)	10	50,00	30,00	20,00	0,00
5	МБОУ "Гимназия № 42" (г. Барнаул)	35	37,14	40,00	22,86	0,00
6	МБОУ "Лицей №129" (г. Барнаул)	28	32,14	35,71	32,14	0,00
7	МБОУ "Гимназия № 27" (г. Барнаул)	13	30,77	46,15	23,08	0,00
8	МБОУ "Лицей "Сигма" (г. Барнаул)	10	30,00	50,00	20,00	0,00
9	МБОУ "Лицей №112" (г. Барнаул)	27	25,93	40,74	33,33	0,00
10	МБОУ "СОШ № 1" (г. Бийск)	16	25,00	25,00	50,00	0,00
11	МБОУ "Гимназия № 11" (г. Бийск)	14	21,43	42,86	35,71	0,00
12	МБОУ "СОШ №127" (г. Барнаул)	16	18,75	56,25	25,00	0,00
13	МБОУ "Гимназия №5" (г. Барнаул)	11	18,18	45,45	36,36	0,00
14	МБОУ "Гимназия №22" (г. Барнаул)	13	15,38	76,92	7,69	0,00
15	МБОУ "Лицей № 6" (г. Рубцовск)	10	10,00	60,00	30,00	0,00
16	МБОУ "Гимназия №85" (г. Барнаул)	13	0,00	46,15	53,85	0,00
17	МБОУ "Лицей № 17" (г. Славгород)	10	0,00	30,00	70,00	0,00

2.4.2. Перечень ОО, продемонстрировавших низкие результаты ЕГЭ по предмету

Таблица 2-12

№ п/п	Наименование ОО	Количество ВТГ, чел.	Доля ВТГ, получивших тестовый балл			
			ниже минимального	от минимального балла до 60 баллов	от 61 до 80 баллов	от 81 до 100 баллов
1	МБОУ "СОШ №18" (г. Бийск)	16	31,25	62,50	6,25	0,00
2	МБОУ "СОШ №113 имени Сергея Семенова" (г. Барнаул)	12	25,00	50,00	25,00	0,00
3	МБОУ СОШ № 3 (г. Бийск)	12	25,00	41,67	25,00	8,33
4	МБОУ "СОШ №114" (г. Барнаул)	14	21,43	35,71	35,71	7,14
5	МБОУ "СОШ № 19 города Новоалтайска Алтайского края" (г. Новоалтайск)	10	20,00	70,00	0,00	10,00
6	МБОУ "СОШ №55" (г. Барнаул)	10	20,00	50,00	30,00	0,00
7	МБОУ "СОШ №118" (г. Барнаул)	11	18,18	45,45	36,36	0,00

№ п/п	Наименование ОО	Количество ВТГ, чел.	Доля ВТГ, получивших тестовый балл			
			ниже минималь- ного	от минималь- ного балла до 60 баллов	от 61 до 80 бал- лов	от 81 до 100 бал- лов
8	МБОУ "СОШ №89" (г. Барнаул)	12	16,67	50,00	25,00	8,33
9	МБОУ "Гимназия №8" (г. Рубцовск)	24	16,67	50,00	33,33	0,00
10	МБОУ "СОШ №102" (г. Барнаул)	12	16,67	25,00	50,00	8,33
11	МАОУ "СОШ №138" (г. Барнаул)	14	14,29	42,86	21,43	21,43
12	МБОУ "Гимназия № 1" (г. Бийск)	21	14,29	42,86	23,81	19,05
13	МБОУ "СОШ №126" (г. Барнаул)	16	12,50	43,75	31,25	12,50
14	МБОУ "Лицей № 121" (г. Барнаул)	19	10,53	31,58	36,84	21,05
15	МБОУ "СОШ №34" (г. Бийск)	10	10,00	70,00	20,00	0,00
16	МАОУ "СОШ №134" (г. Барнаул)	10	10,00	60,00	20,00	10,00
17	МАОУ "СОШ №136" (г. Барнаул)	10	10,00	20,00	50,00	20,00
18	МБОУ "СОШ №107" (г. Барнаул)	10	10,00	20,00	70,00	0,00
19	МБОУ "Гимназия №40" (г. Барнаул)	11	9,09	27,27	54,55	9,09
20	МБОУ "СОШ №125" (г. Барнаул)	24	8,33	20,83	58,33	12,50
21	МБОУ СОШ №12 (г. Бийск)	13	7,69	53,85	30,77	7,69
22	МАОУ "СОШ №137" (г. Барнаул)	13	7,69	38,46	53,85	0,00
23	МБОУ "СОШ №128" (г. Барнаул)	28	7,14	14,29	50,00	28,57
24	МБОУ "Мамонтовская СОШ" (Ма- монтовский район)	15	6,67	66,67	20,00	6,67
25	МБОУ "Гимназия №80" (г. Барнаул)	15	6,67	40,00	40,00	13,33
26	КГБОУ "Бийский лицей-интернат Ал- тайского края» (БЛИАК) (краевые об- разовательные организации)	33	6,06	12,12	54,55	27,27
27	МБОУ "Гимназия №123" (г. Барнаул)	19	5,26	21,05	31,58	42,11
28	МБОУ "Лицей №101" (г. Барнаул)	20	5,00	45,00	40,00	10,00

2.5. ВЫВОДЫ о характере изменения результатов ЕГЭ по предмету

1. Распределение тестовых баллов участников ЕГЭ по информатике в текущем году аппроксимируется к нормальной кривой, но если в 2023 и 2024 годах оно имело выраженную правостороннюю асимметрию (сдвиг), а в 2025 году

этот сдвиг сгладился, то результаты 2026 года показывают уже левостороннюю асимметрию. То есть по этим результатам становится сложнее дифференцировать высокобалльников. Отчасти это объясняется приятным фактом – увеличением высокобалльных работ.

2. Значение среднего балла в 2026 году выросло на 2,26 по сравнению с 2025 годом и на 3,42 по сравнению с 2024 годом, что является положительной характеристикой.

3. Доля участников, не преодолевших минимальный барьер уменьшилось на 1,35% в сравнении с 2024 годом и на 0,93% в сравнении с 2024 годом. Значительно уменьшилась доля участников, набравших от 40 до 60 баллов: на 5,92% в сравнении с 2024 годом и на 6,87% в сравнении с 2024 годом. При этом изменение доли участников, набравших от 60 до 80 баллов, имеет положительную динамику: на 4,57% выше, чем в 2025 году и на 2,55% выше, чем в 2024 году. Доля высокобалльников (от 81 до 100 баллов) выросла за 2 года на 5,24%. Это весьма положительные характеристики.

4. Участников ЕГЭ по информатике, набравших 100 баллов в текущем году 5, в 2024 году это был только 1 человек, в 2025 году в Алтайском крае было 7 100-балльников.

5. Лучшие результаты, как и в 2024-2025 годах, демонстрируют выпускники текущего года, обучающиеся по программам СОО. По категориям ОО это СОО с углубленным изучением информатики, гимназии, лицеи. Выпускники СПО демонстрируют преимущественно низкую базовую подготовку. Выпускники прошлых лет и категория ВПЛ в основной массе либо не преодолели минимальный барьер, либо имеют низкую базовую подготовку, только 11,11 процентов из них получили высокие баллы (от 81 до 100).

6. Доля участников экзамена с базовым уровнем подготовки по информатике более, чем в 2 раза выше, чем доля участников с углубленным уровнем подготовки, при этом, что закономерно, среди второй группы в 2 раза выше доля высокобалльников и в 3 раза ниже доля не преодолевших минимальный барьер.

7. В перечень ОО, продемонстрировавших наиболее высокие результаты ЕГЭ по предмету четыре года подряд включена организация МБОУ «Лицей №124» (г. Барнаул), три года подряд МАОУ «СОШ №132» им. Н.М. Малахова (г. Барнаул) и два года подряд МБОУ «Гимназия №42» (г. Барнаул) и МБОУ «Гимназия №27» (г. Барнаул). После двух лет отсутствия в список ОО с высокими результатами вернулся КГБОУ «АКПЛ», это закономерное событие, потому как эта организация стабильно лидирует по количеству побед школьников на олимпиадах по программированию, по массовости участников ЕГЭ по предмету и до 2024 года ежегодно демонстрировала большую долю участников экзамена с высокими результатами, считается организацией с высоким уровнем физико-математической подготовки.

8. Стабильно низкие результаты демонстрируют следующие ОО: МБОУ «СОШ №118» (г. Барнаул); МБОУ "СОШ №89» (г. Барнаул); МАОУ «СОШ №138» (г. Барнаул); МБОУ «Гимназия №1» (г. Бийск); МБОУ «СОШ №102» (г. Бар-

наул); МБОУ «СОШ №107» (г. Барнаул); МБОУ «СОШ №12» (г. Бийск).

Принимая во внимание, что эти организации и в прошлом году вошли в перечень 5% организаций с наиболее низкими результатами, это должно было быть учтено и исправлено администрацией и учителями информатики СОШ и гимназии.

Резко перешли из таблицы с высокими результатами в таблицу с низкими результатами следующие ОО:

МБОУ «СОШ №113 имени Сергея Семенова» (г. Барнаул): в 2025 году ни одного обучающийся из данной ОО не набрали ниже 40 баллов, а в 2026 году таких участников от школы 25%;

МБОУ «Лицей № 121» (г. Барнаул): в 2025 году ни одного обучающийся из данной ОО не набрали ниже 40 баллов, то в 2026 году таких участников от школы 10,53%;

МБОУ «Гимназия №123» (г. Барнаул): в 2025 году ни одного обучающийся из данной ОО не набрали ниже 40 баллов, то в 2026 году таких участников от школы 5,26%;

наблюдается ухудшение результатов ЕГЭ в КГБОУ «БЛИАК» (краевые образовательные организации) – организация с большой долей участников экзамена, призеров и победителей олимпиад по программированию, в 2026 году 6,06% участников из КГБОУ «БЛИАК» набрали ниже 40 баллов.

По результатам приведенного анализа следует в соответствующих организациях принять меры к выявлению упущенного потенциала в подготовке школьников.

9. В целом, статистические результаты ЕГЭ по информатике удовлетворительны и имеют положительную динамику, однако имеется отрицательная динамика в результатах отдельных ОО.

2.6. Связь динамики результатов ЕГЭ по предмету с принятыми на уровне субъекта Российской Федерации, муниципалитета, образовательных организаций мерами повышения качества освоения ФОП

Динамика результатов ЕГЭ по информатике в Алтайском крае за последние годы демонстрирует опережающий общероссийский тренд роста и напрямую коррелирует с реализацией комплекса управленческих мер федерального, краевого и муниципального уровней. В условиях стратегического приоритета государства на развитие инженерного, технологического и естественно-научного образования регион показывает устойчивое улучшение количественных показателей: при увеличении числа участников (более 1500 тысяч выпускников) средний балл вырос до 57,63 (+2,26), а доля высокобалльников увеличилась 15,02% (+2,7 п.п.). Доля не преодолевших минимальный порог значительно сократилась – на 1,35 п.п. (до 15,9%).

Положительная динамика обусловлена следующими факторами:

переход школ региона на федеральные образовательные программы (ФОП) и утверждение единого федерального перечня цифровых инструментов устранили методический разноречивый. Это позволило синхронизировать глубину изучения алгоритмизации, программирования на языках Python и C++, элементов теории графов и систем счисления, исключив ситуацию, когда выпускники из разных районов имели несопоставимый объем знаний для решения заданий второй части;

системная интеграция Единого банка заданий и открытого банка КИМ в урочную деятельность позволила учителям сместить фокус с формального прохождения тем на отработку конкретных прототипов задач повышенной сложности (рекурсивные алгоритмы, динамическое программирование, анализ информационных моделей). Использование тренажеров автоматической проверки кода обеспечило прозрачность требований экзаменационной модели и рост среднего балла за задания высокого уровня сложности;

инициированная Президентом Российской Федерации возможность пересдать один предмет до конца приемной кампании стала важным фактором снижения стресса: из 258 участников пересдач по информатике 212 (82,2%) повысили свой результат, 25 (9,7%) – показали стабильные результаты;

масштабирование проекта РЭШ обеспечило доступ к видеоурокам от ведущих преподавателей МФТИ, МГУ и педагогических университетов строго в логике ФОП. Для информатики это стало критически важным инструментом выравнивания подготовки в малокомплектных школах отдаленных сел, где ранее отсутствовали профильные учителя;

реализация региональных программ материального стимулирования помогла привлечь квалифицированных учителей информатики в сельские территории. Целевые выплаты педагогам, переезжающим работать в Ключевской, Михайловский районы, позволили повысить там средний балл благодаря внедрению проектной деятельности;

система наставничества: закрепление за молодыми учителями информатики опытных экспертов-предметников краевого учебно-методического объединения предотвратило типичные ошибки начинающих педагогов при подготовке к оформлению блок-схем, обоснованию эффективности алгоритмов и работе с системами управления базами данных (СУБД);

проведение массовых пробных тестирований в формате ЕГЭ с внешней проверкой работ экспертами АлтГТУ, АлтГПУ и АГУ позволило выявить системные затруднения в написании программного кода еще в первом полугодии 11 класса;

выравнивание доступа к ресурсам: внедрение сетевых форматов обучения, где обучающиеся из районов края получили дистанционный доступ к сильным педагогическим кадрам краевого центра и специализированному лицензионному ПО (включая среды разработки IDE и виртуальные лаборатории);

сетевое взаимодействие: реализация курсов по выбору на базе Алтайского государственного технического университета им. И.И. Ползунова, Алтайского государственного университета, Бийского технологического лицея и Рубцовского аграрно-промышленного техникума создала условия для ранней профориентации. Школьники получили возможность углубленно изучать прикладные разделы информатики (машинное обучение, кибербезопасность, робототехнику, веб-разработку) непосредственно на базе будущих мест обучения;

адресная работа с ОО зоны риска: в перечень школ с низкими результатами в 2025 году попали учреждения Ельцовского, Калманского, Кытмановского, Смоленского, Третьяковского, Шипуновского районов. Динамика снижения доли не преодолевших минимум свидетельствует об эффективности работы региональных методических служб;

изменение структуры выбора предмета: рост среднего балла (+2,26) частично объясняется изменением состава участников. Благодаря популяризации IT-специальностей произошел качественный отбор: слабые обучающиеся не выбирают предмет для сдачи, остались наиболее мотивированные участники.

Комплекс принятых мер привел к качественному изменению структуры результатов: сократилась доля выпускников, чьи баллы находятся в «провальной» зоне, и выросла группа обучающихся, набравших свыше 81 балла. Это соответствует потребностям региональной экономики в инженерных и IT-кадрах для предприятий агропромышленного комплекса, оборонно-промышленного кластера и IT-сферы края.

Раздел 3. МЕТОДИЧЕСКИЙ АНАЛИЗ РЕЗУЛЬТАТОВ ВЫПОЛНЕНИЯ ЗАДАНИЙ КИМ И РЕКОМЕНДАЦИИ ДЛЯ РЕГИОНАЛЬНОЙ СИСТЕМЫ ОБРАЗОВАНИЯ

3.1. Статистический анализ выполнения заданий КИМ

3.1.1. Основные статистические характеристики выполнения заданий КИМ в 2026 году

Основные статистические характеристики выполнения заданий в целом представлены в Таб.2-13. Информация о результатах оценивания выполнения заданий, в том числе в разрезе данных о получении того или иного балла по критерию оценивания выполнения каждого задания КИМ представлена в Таб. 2-14.

Таблица 2-13

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в субъекте Российской Федерации ¹ в группах участников экзамена с разными уровнями подготовки				
			средний, %	в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
Задания с кратким ответом							
1.	Умение представлять и считать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)	Базовый	92,49	65,22	96,04	97,69	99,16
2.	Умение строить таблицы истинности и логические схемы	Базовый	78,04	22,17	77,3	93,85	99,16
3.	Умение поиска информации в реляционных базах данных	Базовый	69,82	28,26	68,83	79,04	92,05
4.	Умение кодировать и декодировать информацию	Базовый	74,55	32,61	72,97	85,96	93,72

¹ Вычисляется по формуле $p = \frac{N}{nm} \cdot 100\%$, где N – сумма первичных баллов, полученных всеми участниками группы за выполнение задания, n – количество участников в группе, m – максимальный первичный балл за задание.

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в субъекте Российской Федерации ¹ в группах участников экзамена с разными уровнями подготовки				
			средний, %	в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
5.	Формальное исполнение простого алгоритма, записанного на естественном языке, или умение создавать линейный алгоритм для формального исполнителя с ограниченным набором команд, или умение восстанавливать исходные данные линейного алгоритма по результатам его работы	Базовый	55,18	3,91	36,4	78,85	96,65
6.	Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов	Базовый	63,15	8,26	54,41	82,31	94,56
7.	Умение определять объём памяти, необходимый для хранения графической и звуковой информации	Базовый	74,81	20	71,17	91,54	99,58
8.	Кодирование данных, комбинаторика, системы счисления.	Базовый	55,25	2,17	32,43	83,46	97,91
9.	Умение обрабатывать числовую информацию в электронных таблицах	Базовый	44,88	2,17	17,48	69,62	95,82
10.	Информационный поиск средствами текстового процессора	Базовый	79,47	54,35	76,76	87,5	92,47

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в субъекте Российской Федерации ¹ в группах участников экзамена с разными уровнями подготовки				
			средний, %	в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
11.	Умение подсчитывать информационный объём сообщения	Повышенный	41	3,04	23,78	54,81	87,45
12.	Умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд	Повышенный	65,16	6,09	54,95	87,69	96,65
13.	Умение использовать маску подсети	Повышенный	52,72	5,65	35,5	71,92	96,23
14.	Знание позиционных систем счисления	Повышенный	43,98	1,3	15,68	70,58	92,89
15.	Знание основных понятий и законов математической логики	Повышенный	34,33	1,74	10,45	50	87,03
16.	Вычисление рекуррентных выражений	Повышенный	58,74	3,48	37,66	87,12	99,16
17.	Умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования	Повышенный	32,58	0	4,14	48,65	94,98
18.	Умение использовать электронные таблицы для обработки целочисленных данных	Повышенный	56,41	7,39	42,7	75,96	92,89
19.	Умение анализировать алгоритм логической игры	Базовый	60,04	10,43	42,7	83,46	97,07

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в субъекте Российской Федерации ¹ в группах участников экзамена с разными уровнями подготовки				
			средний, %	в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
20.	Умение найти выигрышную стратегию игры	Повышенный	50,97	0,43	26,13	78,46	97,49
21.	Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию	Высокий	51,81	1,3	26,67	79,81	97,91
22.	Построение математических моделей для решения практических задач. Архитектура современных компьютеров. Многопроцессорные системы.	Повышенный	41,45	5,22	20,72	56,15	92,47
23.	Умение анализировать ход исполнения алгоритма	Повышенный	26,3	0	3,24	34,42	87,45
24.	Умение создавать собственные программы (10–20 строк) для обработки символьной информации	Высокий	2,91	0	0,36	0,96	15,9
25.	Умение создавать собственные программы (10–20 строк) для обработки целочисленной информации	Высокий	12,18	0	0	8,27	60,67
26.	Умение обрабатывать целочисленную информацию с использованием сортировки	Высокий	4,08	0	0,09	0,96	24,06

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в субъекте Российской Федерации ¹ в группах участников экзамена с разными уровнями подготовки				
			средний, %	в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
27.	Умение выполнять последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и построение модели, преобразование данных, визуализация данных, интерпретация результатов	Высокий	10,33	0	0,18	5,77	53,77

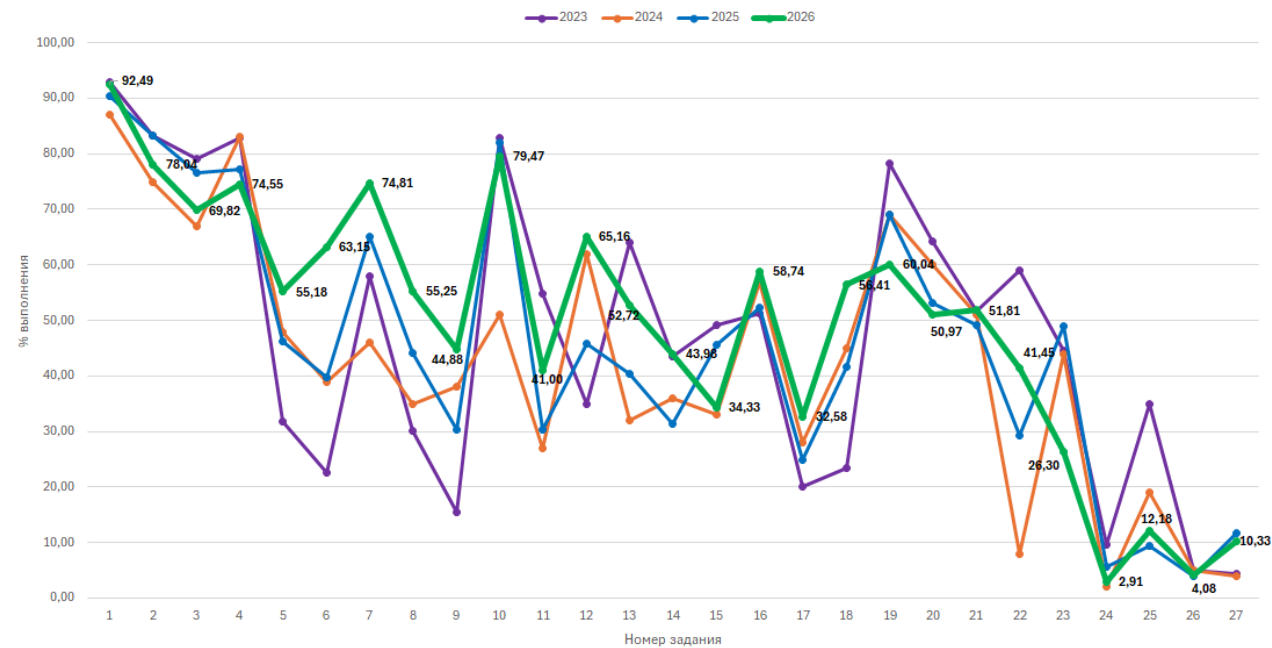


Рисунок 2-4. Динамика результатов выполнения заданий участниками ЕГЭ по информатике в 2023-2026 г.г.

Таблица 2-14

Номер задания / критерия оценивания в КИМ	Количество полученных первичных баллов	Процент участников экзамена в субъекте Российской Федерации, получивших соответствующий первичный балл за выполнения задания в группах участников экзамена с разными уровнями подготовки			
		в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б., %	в группе от 61 до 80 т.б., %	в группе от 81 до 100 т.б., %
Количество участников в группе		230	555	520	239
1	X	1,74	0,18	0,00	0,00
	0	33,04	3,78	2,31	0,84
	1	65,22	96,04	97,69	99,16
2	X	6,52	1,80	0,00	0,00
	0	71,30	20,90	6,15	0,84
	1	22,17	77,30	93,85	99,16
3	X	6,09	1,26	0,38	0,00
	0	65,65	29,91	20,58	7,95
	1	28,26	68,83	79,04	92,05
4	X	3,91	0,54	0,00	0,00
	0	63,48	26,49	14,04	6,28
	1	32,61	72,97	85,96	93,72
5	X	33,91	24,50	4,81	0,00
	0	62,17	39,10	16,35	3,35
	1	3,91	36,40	78,85	96,65
6	X	18,70	5,77	0,58	0,00
	0	73,04	39,82	17,12	5,44
	1	8,26	54,41	82,31	94,56
7	X	13,48	3,24	0,19	0,00
	0	66,52	25,59	8,27	0,42
	1	20,00	71,17	91,54	99,58
8	X	25,22	19,10	2,12	0,00
	0	72,61	48,47	14,42	2,09
	1	2,17	32,43	83,46	97,91
9	X	46,96	47,21	12,12	0,00

	0	50,87	35,32	18,27	4,18
	1	2,17	17,48	69,62	95,82
10	X	2,17	0,54	0,00	0,00
	0	43,48	22,70	12,50	7,53
	1	54,35	76,76	87,50	92,47
11	X	26,52	11,89	1,35	0,00
	0	70,43	64,32	43,85	12,55
	1	3,04	23,78	54,81	87,45
12	X	46,52	19,28	2,31	0,00
	0	47,39	25,77	10,00	3,35
	1	6,09	54,95	87,69	96,65
13	X	29,57	20,36	5,77	0,00
	0	64,78	44,14	22,31	3,77
	1	5,65	35,50	71,92	96,23
14	X	49,13	41,80	7,50	0,00
	0	49,57	42,52	21,92	7,11
	1	1,30	15,68	70,58	92,89
15	X	47,83	44,68	13,08	0,42
	0	50,43	44,86	36,92	12,55
	1	1,74	10,45	50,00	87,03
16	X	47,83	31,71	5,19	0,42
	0	48,70	30,63	7,69	0,42
	1	3,48	37,66	87,12	99,16
17	X	68,70	71,17	26,35	0,00
	0	31,30	24,68	25,00	5,02
	1	0,00	4,14	48,65	94,98
18	X	49,13	18,56	2,88	0,00
	0	43,48	38,74	21,15	7,11
	1	7,39	42,70	75,96	92,89
19	X	31,74	15,86	0,96	0,00
	0	57,83	41,44	15,58	2,93
	1	10,43	42,70	83,46	97,07
20	X	47,39	27,21	2,12	0,00

	0	52,17	46,67	19,42	2,51
	1	0,43	26,13	78,46	97,49
21	X	46,09	32,07	4,04	0,00
	0	52,61	41,26	16,15	2,09
	1	1,30	26,67	79,81	97,91
22	X	50,00	39,46	12,88	1,67
	0	44,78	39,82	30,96	5,86
	1	5,22	20,72	56,15	92,47
23	X	53,48	53,69	28,27	3,35
	0	46,52	43,06	37,31	9,21
	1	0,00	3,24	34,42	87,45
24	X	66,09	77,12	70,19	35,15
	0	33,91	22,52	28,85	48,95
	1	0,00	0,36	0,96	15,90
25	X	86,52	94,05	75,77	17,15
	0	13,48	5,95	15,96	22,18
	1	0,00	0,00	8,27	60,67
26	X	79,57	86,85	81,15	43,93
	0	20,43	12,97	17,12	24,27
	1	0,00	0,18	1,54	15,48
	2	0,00	0,00	0,19	16,32
27	X	84,35	90,81	80,77	26,36
	0	15,65	8,83	11,15	10,88
	1	0,00	0,36	4,62	17,99
	2	0,00	0,00	3,46	44,77

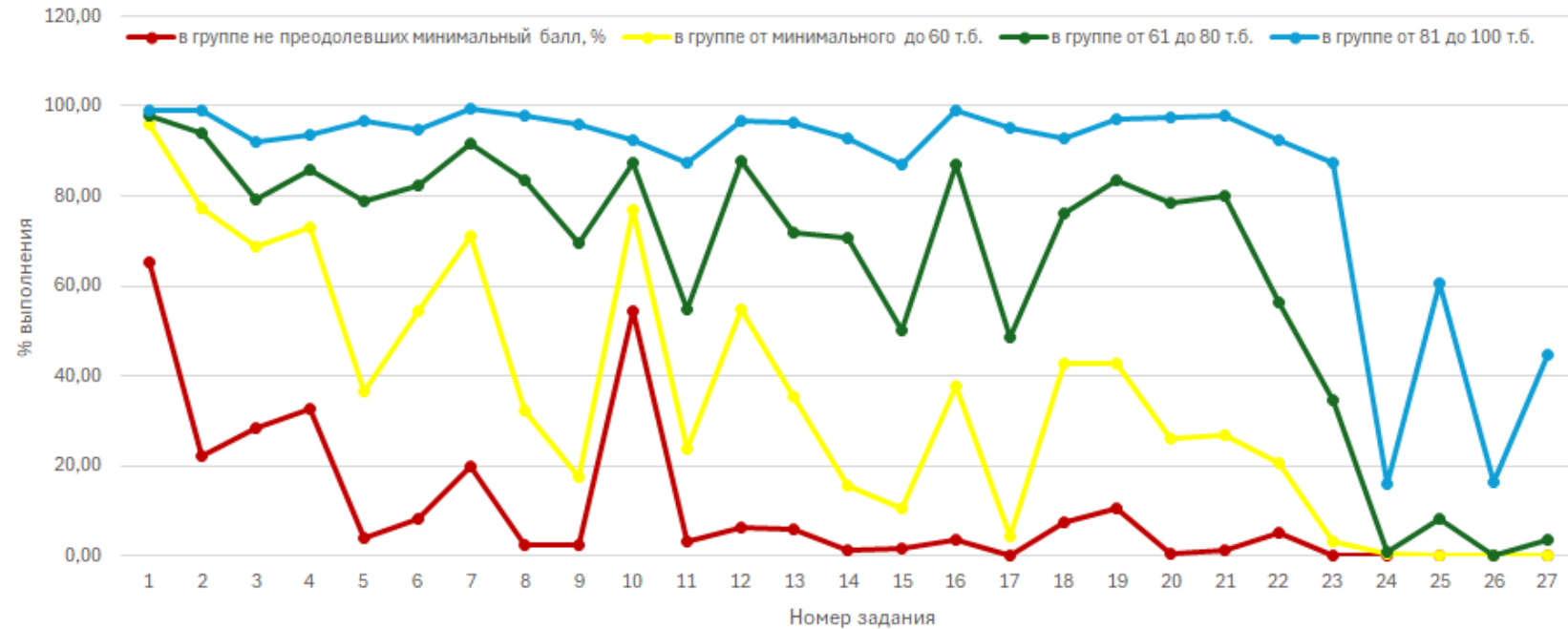


Рисунок 2-5. Выполнение заданий ЕГЭ 2025 г. участниками с разными уровнями подготовки

Выявление для каждой группы участников ЕГЭ заданий, выполняемых с наибольшей и наименьшей успешностью

Таблица 2-15

	в группе не преодолевших ми- нимальный балл	в группе от мини- мального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
Наиболее успешно выполненные за- дания базового уровня сложности	Задания 1, 10	Задания 1, 2, 3, 4, 7, 10		
Наименее успешно выполненные за- дания базового уровня сложности	Задания 5, 6, 8, 9	Задание 9	Задание 9	
Наиболее успешно выполненные за- дания повышенного уровня сложно- сти	—	Задания 12, 18	Задания 12, 13, 14, 16, 18, 20	Задания 11, 12, 13, 14, 15, 16, 17, 18, 20, 22, 23

	в группе не преодолевших ми- нимальный балл	в группе от мини- мального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
Наименее успешно выполненные за- дания повышенного уровня сложно- сти		Задания 14, 15, 17, 23	Задание 23	–
Наиболее успешно выполненные за- дания высокого уровня сложности		Задание 21	Задание 21	Задания 21
Наименее успешно выполненные за- дания высокого уровня сложности			Задания 24, 25, 26, 27	Задания 24, 26

3.1.2. Прочие результаты статистического анализа (при наличии)

При сравнении результатов КЕГЭ 2026 года и предыдущих лет выявляется, что повысились показатели выполнимости заданий 5, 6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, 21, 22. Остались на прежнем уровне показатели выполнимости заданий 1, 10, 20, 24, 25, 26, 27. Понижились результаты выполнения заданий 2, 3, 4, 15, 19, 23.

В группе высокобалльников (от 81 до 100) кривая результатов существенно сглажена к более высоким показателям (рис. 2-5), остались наиболее заметные «пики» вниз с показателями результатов выполнения заданий 11 и 15 повышенного уровня сложности и с показателями результатов выполнения заданий 24 и 26 высокого уровня сложности.

В группе участников, набравших от 60 до 80 баллов, значительно улучшены результаты выполнения задания 6, 14 и 22, а вот задания 9, 11 и 17 остались как и в предыдущие годы проблемными для экзаменуемых, при этом значительно ухудшились результаты выполнения заданий 15 и 23.

В группе участников, набравших от 40 до 60 баллов, улучшились показатели выполнения заданий 5, 6, 12, при этом понизились результаты выполнения заданий 19, 20, 23. Остальные показатели в этой группе остались примерно на прежнем уровне.

В группе участников экзамена, не преодолевших минимальный барьер, значительных изменений нет.

Таким образом, можно выделить самые «решаемые» задания во всех группах участников и самые «проблемные» задания, при решении которых допущено больше всего ошибок или к их решению вовсе не приступали:

наиболее решаемые задания – 1, 10, 12, 21;

самые проблемные задания – 9, 11, 15, 23, 24, 26.

3.2. Методический анализ выполнения заданий КИМ обучающимися с разными уровнями подготовки и реко-

мендации для учителей по совершенствованию преподавания учебного предмета

3.2.1. Методический анализ выполнения заданий обучающимися с минимальным уровнем подготовки (не преодолевших минимальный балл), включая методические рекомендации для учителей

3.2.1.1. Описание предметной подготовки участников ЕГЭ с минимальным уровнем подготовки, анализ типичных дефицитов в подготовке

○ *Описание достигнутых предметных результатов ФГОС: освоенных тем программы, сформированных умений и т.п. (если имеются)*

Группа участников, не преодолевших минимальный балл, показала наиболее высокие результаты при выполнении заданий 1 и 10 базового уровня сложности, по остальным заданиям очень низкие результаты.

Можно констатировать, что обучающимися этой группы на высоком уровне освоены следующие элементы содержания школьной программы по информатике и умения:

1) Разделы/темы школьной программы: 11 кл., п. 113.7.2. Представление результатов моделирования в виде, удобном для восприятия человеком. Графическое представление данных (схемы, таблицы, графики). Элементы содержания: Модели и моделирование. Цели моделирования. Адекватность модели моделируемому объекту или процессу. Формализация прикладных задач. Представление результатов моделирования в виде, удобном для восприятия человеком. Графическое представление данных (схемы, таблицы, графики). Умения и навыки: умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде. Метапредметные результаты: Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления.

2) Разделы/темы школьной программы: 10 кл., п. 113.6.3. Текстовый процессор. Средства поиска и автозамены в текстовом процессоре. Элементы содержания: Текстовый процессор. Средства поиска и автозамены в текстовом процессоре. Структурированные текстовые документы. Сноски, оглавление. Правила цитирования источников и оформления библиографических ссылок. Умения и навыки: владение навыками работы с операционными системами и

основными видами программного обеспечения для решения учебных задач по выбранной специализации. Метапредметные результаты: Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.

○ *Анализ типичных дефицитов в предметной подготовке: неосвоенные или слабо освоенные темы программы, несформированные или недостаточно сформированные умения, навыки, виды познавательной деятельности, проявляющиеся в типичных ошибках при выполнении соответствующих заданий*

Прежде всего для данной группы участников стоит задача освоения базового уровня подготовки по информатике.

Таблица 2-16

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП
1.	Задание 2 <u>Элементы содержания</u>: Алгебра логики. Логические операции. Таблицы истинности. Логические выражения. Логические тождества. <u>Умения и навыки</u>: Умение строить логическое выражение в дизъюнктивной и конъюнктивной нормальных формах по заданной таблице истинности; исследовать область истинности высказывания, содержащего переменные; решать несложные логические уравнения. Умение строить таблицы истинности и логические схемы. <u>Метапредметные результаты</u>: <i>Базовые логические действия</i> (Устанавливать существенный признак или основания для сравнения, классификации и обобщения. Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности.). <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях.).	Базовый уровень: 10 кл., п. 113.6.2. Углубленный уровень: 10 кл., п. 114.6.2

2.	Задание 3 <u>Элементы содержания:</u> Табличные (реляционные) базы данных. Поле, запись. Ключ таблицы. Работа с готовой базой данных. Поиск, сортировка и фильтрация данных. Запросы на выборку данных. Многотабличные базы данных. Типы связей между таблицами. <u>Умения и навыки:</u> Владение основными сведениями о базах данных, их структуре, средствах создания и работы с ними. Умение поиска информации в реляционных базах данных. <u>Метапредметные результаты:</u> Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления.	Базовый уровень: 11 кл., п. 113.7.4. Углубленный уровень: 11 кл., п. 114.7.3
3.	Задание 4 <u>Элементы содержания:</u> Двоичное кодирование. Равномерные и неравномерные коды. Декодирование сообщений, записанных с помощью неравномерных кодов. Условие Фано. Построение однозначно декодируемых кодов с помощью дерева. <u>Умения и навыки:</u> Умение кодировать и декодировать информацию. Умение строить код, обеспечивающий наименьшую возможную среднюю длину сообщения при известной частоте символов. <u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской деятельности. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях.)	Базовый уровень: 10 кл., п. 113.6.2. Углубленный уровень: 10 кл., п. 114.6.2
4.	Задание 5 <u>Элементы содержания:</u> Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат. <u>Умения и навыки:</u> Умение анализировать алгоритмы с использованием таблиц трассировки; определять без использования компьютера результаты выполнения несложных программ, включающих циклы, ветвления и подпрограммы, при заданных исходных данных. <u>Метапредметные результаты:</u> <i>Базовые логические действия</i> (Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности). <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской деятельности. Выявлять причинно-следственные связи и актуализиро-	Базовый уровень: 11 кл., п. 113.7.3. Углубленный уровень: 10 кл., п. 114.6.3

	вать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду.)	
5.	Задание 6 <u>Элементы содержания:</u> Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат. <u>Умения и навыки:</u> Умение анализировать алгоритмы с использованием таблиц трассировки; определять без использования компьютера результаты выполнения несложных программ, включающих циклы, ветвления и подпрограммы, при заданных исходных данных. <u>Метапредметные результаты:</u> <i>Базовые логические действия</i> (Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности). <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской деятельности. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду.)	Базовый уровень: 11 кл., п. 113.7.3 Углубленный уровень: 10 кл., п. 114.6.3
6.	Задание 7 <u>Элементы содержания:</u> Кодирование изображений. Оценка информационного объёма графических данных при заданных разрешении и глубине кодирования цвета. Цветовые модели. Кодирование звука. Оценка информационного объёма звуковых данных при заданных частоте дискретизации и разрядности кодирования. Скорость передачи данных. <u>Умения и навыки:</u> Умение определять информационный объём текстовых, графических и звуковых данных при заданных параметрах дискретизации. Умение определять среднюю скорость передачи данных, оценивать изменение времени передачи при изменении информационного объёма данных и характеристик канала связи.	Базовый уровень: 10 кл., п. 113.6.2. Углубленный уровень: 10 кл., п. 114.6.2

	<u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской деятельности. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду.)	
7.	Задание 8 <u>Элементы содержания:</u> Теоретические подходы к оценке количества информации. Единицы измерения количества информации. Алфавитный подход к оценке количества информации. Элементы комбинаторики <u>Умения и навыки:</u> Знание основных понятий и методов, используемых при измерении количества информации. Понимание основных принципов дискретизации различных видов информации. <u>Метапредметные результаты:</u> <i>Базовые логические действия</i> (Устанавливать существенный признак или основания для сравнения, классификации и обобщения). <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской деятельности.)	Базовый уровень: 10 кл., п. 113.6.2 Углубленный уровень: 10 кл., п. 114.6.2
8.	Задание 9 <u>Элементы содержания:</u> Анализ данных с помощью электронных таблиц. Вычисление суммы, среднего арифметического, наибольшего (наименьшего) значения диапазона. <u>Умения и навыки:</u> Умение использовать электронные таблицы для анализа, представления и обработки данных (включая выбор оптимального решения, подбор линии тренда, решение задач прогнозирования); умение использовать табличные (реляционные) базы данных и справочные системы. <u>Метапредметные результаты:</u> <i>Работа с информацией</i> (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.)	Базовый уровень: 11 кл., п. 113.7.4 Углубленный уровень: 10 кл., п. 114.6.4

○ *Реалистичные «зоны роста» в части предметной подготовки для уверенного преодоления минимального балла ЕГЭ*

Основываясь на анализе результатов выполнения заданий в группах участников экзамена 2026 года и сопостав-

лении этих результатов с результатами прошлых лет, полагаем, что не использован группой участников, не преодолевших минимальный балл, потенциал заданий 2, 3, 4, 7 для получения минимального положительного балла.

Ключевым критерием отбора заданий для «зон роста» является сравнение результатов выполнения в группе «ниже минимального балла» с результатами группы «от минимального до 60 т.б.». Если задание показывает значительный разрыв между этими группами, это свидетельствует о высоком потенциале его освоения при целенаправленной подготовке.

Таким образом, для участников ЕГЭ с минимальным уровнем подготовки наиболее реалистично для преодоления минимального барьера акцентировать внимание на следующих элементах содержания и умениях.

Задание 2 (Базовый уровень)

Тема: Умение строить таблицы истинности и логические схемы.

Показатель	Значение
% выполнения в группе «ниже мин.»	22,17%
% выполнения в группе «от мин. до 60»	77,30%
Разрыв между группами	55,13%.

Почему это «зона роста»: колоссальный разрыв между группами (более 55 п.п.) свидетельствует о том, что задание принципиально решаемое, но требует освоения базового алгоритма построения таблиц истинности. Обучающиеся, набравшие минимальный балл, выполняют это задание почти в 4 раза лучше, что подтверждает высокий потенциал роста для слабой группы. Ошибки возникают из-за незнания порядка выполнения логических операций или неумения правильно заполнять таблицу.

Что отрабатывать:

порядок выполнения логических операций: $\neg \rightarrow \wedge \rightarrow \vee \rightarrow \rightarrow \rightarrow \leftrightarrow$ (инверсия, конъюнкция, дизъюнкция, импликация, эквивалентность);

построение таблиц истинности для 2-3 переменных;

использование метода подстановки значений вместо сложных преобразований;

решение задач на поиск значений переменных, при которых логическое выражение истинно.

Задание 3 (Базовый уровень)

Тема: Умение поиска информации в реляционных базах данных.

Показатель	Значение
% выполнения в группе «ниже мин.»	28,26%
% выполнения в группе «от мин. до 60»	68,83%
Разрыв между группами	40,57%

Почему это «зона роста»: разрыв между группами составляет более 40 п.п. Задание проверяет практический навык работы с табличными данными. Обучающиеся, преодолевшие минимальный барьер, выполняют задание почти в 2,5 раза лучше. Это доказывает, что при освоении стандартных приёмов фильтрации и сортировки задание становится доступным для большинства обучающихся.

Что отрабатывать:

поиск информации в одной таблице с использованием условий (фильтрация);
поиск информации в двух связанных таблицах (использование ключевых полей);
сортировка данных по заданному полю (по возрастанию/убыванию);
работа с готовой базой данных в электронных таблицах.

Задание 4 (Базовый уровень)

Тема: Умение кодировать и декодировать информацию (условие Фано).

Показатель	Значение
% выполнения в группе «ниже мин.»	32,61%
% выполнения в группе «от мин. до 60»	72,97%
Разрыв между группами	40,36%

Почему это «зона роста»: разрыв между группами составляет более 40 п.п. Задание имеет наглядный алгоритм построения дерева кодов. Ошибки возникают из-за непонимания условия Фано (прямого и обратного) или неумения строить дерево. При освоении методики построения дерева задача решается быстро и надёжно. Обучающиеся из группы «от мин. до 60» демонстрируют результат почти в 2,5 раза выше.

Что отрабатывать:

построение двоичного дерева для заданных частот символов;
проверка условия Фано: никакой код не должен быть началом другого кода;
определение минимальной длины кода для заданных частот;
решение задач на нахождение кодов, удовлетворяющих условию Фано.

Задание 7 (Базовый уровень)

Тема: Умение определять объём памяти для хранения графической и звуковой информации.

Показатель	Значение
% выполнения в группе «ниже мин.»	20,00%
% выполнения в группе «от мин. до 60»	71,17%
Разрыв между группами	51,17%

Почему это «зона роста»: задание показывает колоссальный разрыв между группами (более 51 п.п.). Обучающиеся, преодолевшие минимальный барьер, выполняют задание в 3,5 раза лучше. Это означает, что при целенаправленной отработке данное задание может быть освоено практически всеми учащимися. Формулы расчёта объёма информации стандартны и легко запоминаются. Именно это задание имеет самый высокий потенциал роста в данной группе.

Что отрабатывать:

формула для графических данных: $V = \text{ширина} \times \text{высота} \times \text{глубина цвета (в битах)}$;

формула для звуковых данных: $V = \text{частота} \times \text{время} \times \text{разрядность (в битах)}$;

перевод единиц измерения: биты $\rightarrow$ байты $\rightarrow$ Кбайты $\rightarrow$ Мбайты (деление на 8, 1024);

решение задач на скорость передачи данных: $V = \text{время} \times \text{скорость}$.

Задание 12 (Повышенный уровень) – РЕЗЕРВНОЕ

Тема: Умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд.

Показатель	Значение
% выполнения в группе «ниже мин.»	6,09%
% выполнения в группе «от мин. до 60»	54,95%
Разрыв между группами	48,86%

Почему это «зона роста»: несмотря на низкий результат в слабой группе (6,09%), задание показывает огромный потенциал роста (разрыв почти 49 п.п.). Обучающиеся из группы «от мин. до 60» выполняют его более чем в 9 раз лучше. При целенаправленной подготовке может быть освоено как дополнительное.

Что отрабатывать:

выполнение «ручной прокрутки» (трассировки) алгоритма;

пошаговая фиксация состояния исполнителя;

моделирование движения исполнителя по командам.

Итоговый список элементов содержания и умений (приоритетные темы для преодоления минимального балла, включая успешно освоенные)

№ задания	Проверяемые элементы содержания	Ключевые умения и навыки
1	Представление данных в разных типах информационных моделей (схемы, карты, таблицы, графики).	Чтение графов и схем. Поиск путей в графе. Определение количества путей между пунктами.
2	Алгебра логики. Логические операции. Таблицы истинности.	Построение таблиц истинности для 2-3 переменных. Определение порядка выполнения логических операций. Поиск значений переменных, при которых выражение истинно.
3	Поиск информации в реляционных базах данных.	Фильтрация данных по условиям. Сортировка по заданному полю. Поиск в двух связанных таблицах. Использование ключевых полей.
4	Двоичное кодирование. Равномерные и неравномерные коды. Условие Фано.	Построение дерева кодов. Проверка прямого и обратного условия Фано. Определение минимальной длины кода.
7	Кодирование изображений и звука. Оценка информационного объёма.	Расчёт объёма графических данных по формуле $V = \text{ширина} \times \text{высота} \times \text{глубина цвета}$. Расчёт объёма звуковых данных по формуле $V = \text{частота} \times \text{время} \times \text{разрядность}$. Перевод единиц измерения (биты $\rightarrow$ байты $\rightarrow$ Кбайты $\rightarrow$ Мбайты).
10	Информационный поиск средствами текстового процессора.	Поиск и замена текста. Работа с масками и шаблонами. Определение пути к файлу.
12	Исполнение алгоритма для конкретного исполнителя с фиксированным набором команд.	Трассировка алгоритма. Моделирование движения исполнителя.

ВЫВОД

Применение знаний по успешно освоенным темам (задания 1 и 10) и ликвидация дефицитов знаний по выделенным выше темам (задания 2, 3, 4, 7) позволяют выполнить 6 заданий для преодоления минимального балла.

Рекомендуемая последовательность освоения:

Этап	Задания	Содержание работы
1 этап: Закрепление	№ 1, 10	Отработка навыков чтения графов и поиска информации в текстовых документах.
2 этап: Освоение	№ 7	Изучение формул для расчёта объёма информации. Отработка перевода единиц измерения.
3 этап: Освоение	№ 4	Построение дерева кодов. Проверка условия Фано. Задание имеет наглядный алгоритм.
4 этап: Освоение	№ 2	Построение таблиц истинности. Работа с логическими операциями.
5 этап: Освоение	№ 3	Фильтрация и сортировка данных в базах данных. Работа с двумя таблицами.

Этап	Задания	Содержание работы
6 этап: Резерв	№ 12	Трассировка алгоритмов для исполнителей. Задание повышенного уровня, может быть использовано как дополнительное.

Ключевые методические рекомендации:

Акцент на практику: Все задания имеют стандартные алгоритмы решения, которые необходимо многократно отработать на тренировочных задачах.

Пошаговый подход: Разбивать сложные задачи на простые шаги (особенно для заданий 2, 4, 7).

Визуализация: Использовать схемы и таблицы при решении (деревья кодов, таблицы истинности).

Контроль единиц измерения: Особое внимание при решении задания 7 уделять переводу битов в байты и килобайты.

Регулярное повторение: Включать задания из «зон роста» в каждое занятие для формирования устойчивых навыков.

Использование эталонных решений: Показывать учащимся образцы оформления решений для заданий 2, 3, 4, 7, чтобы они могли действовать по алгоритму.

3.2.1.2. Методический анализ качества освоения метапредметных результатов ФГОС участниками ЕГЭ с минимальным уровнем подготовки

○ *Развернутый комментарий на основе заданий / групп заданий, на успешность выполнения которых могла повлиять достаточная или недостаточная сформированность метапредметных умений, с указанием соответствующих метапредметных умений и их проявления в типичных ошибках в ответах участников экзамена на соответствующие задания*

При успешном выполнении заданий 1 и 10 участники ЕГЭ с минимальным уровнем подготовки продемонстрировали достаточную сформированность следующих метапредметных результатов: Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления (таблицы и графы). Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач (при работе с текстовыми редакторами).

Недостаточная сформированность метапредметных умений показана по группам УУД:

Базовые логические действия (Устанавливать существенный признак или основания для сравнения, классификации и обобщения. Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности.).

Базовые исследовательские действия (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду.).

Работа с информацией (Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач (при работе с табличными редакторами).

Рассмотрим типичные ошибки в ответах участников экзамена на примере конкретных заданий базового уровня сложности, которые выполнены анализируемой группой наименее успешно, и продемонстрируем влияние недостаточной сформированности соответствующих метапредметных умений.

Задание 5

Тема: Выполнение и анализ простых алгоритмов.

Уровень сложности: базовый.

Рекомендуемое время выполнения: 4 минуты.

Проверяется умение формально исполнять простой алгоритм, записанный на естественном языке, или умение создавать линейный алгоритм для формального исполнителя с ограниченным набором команд, или умение восстанавливать исходные данные линейного алгоритма по результатам его работы.

Проверяемые элементы содержания: Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат.

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Умение анализировать алгоритмы с использованием таблиц трассировки; определять без использования компьютера результаты выполнения несложных программ, включающих циклы, ветвления и подпрограммы, при заданных исходных данных.

Проверяемые метапредметные результаты: Базовые логические действия. Базовые исследовательские действия.

Что нужно знать:

сумма двух цифр в десятичной системе счисления находится в диапазоне от 0 до 18 ($9+9$);

в некоторых задачах нужно иметь представление о системах счисления (могут использоваться цифры восьмеричной, шестнадцатеричной и других систем счисления);

бит чётности – это дополнительный контрольный бит, который добавляется к двоичному коду так, чтобы количество единиц в полученном двоичном коде стало чётным; если в исходном коде уже было чётное количество единиц, дописывается 0, если нечётное – дописывается 1;

при добавлении к записи числа нуля справа число увеличивается в q раз, где q – это основание системы счисления;

чтобы отбросить последнюю цифру в записи числа, нужно разделить число на основание системы счисления нацело (остаток отбрасывается).

Пример формулировки задания (используется открытый вариант №307)

На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N .

2. Далее эта запись обрабатывается по следующему правилу:

а) если число N чётное, то к этой записи справа и слева дописываются по две единицы;

б) если число N нечётное, то в конец двоичной записи (справа) дописываются два нуля, а в начало (слева) дописывается единица.

Полученная таким образом запись (в ней на три или четыре разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R .

3. Результат переводится в десятичную систему и выводится на экран.

Например, для исходного числа $1310 = 11012$ результатом является число $11101002 = 11610$, а для исходного числа $1410 = 11102$ это число $111110112 = 25110$.

Укажите наименьшее число R , превышающее 119, которое может быть результатом работы данного алгоритма. В ответе запишите это число в десятичной системе счисления.

Решение (Вариант 1)

Программирующий обучающийся вполне может реализовать перебор чисел и проверку выполнения условий для преобразованных значений по описанному алгоритму с применением языка программирования.

Пример программы на языке Python:

```
def solve():# Перебираем N, начиная с 1
    N = 1
    results = []
    # Будем искать, пока не найдем достаточно много результатов > 119
    # или пока не убедимся, что нашли минимальный
    found = []
    # Ограничим перебор разумным пределом, так как N растет и R тоже
    # Можно перебирать, пока не найдем 20 чисел > 119 и возьмем минимум
    while len(found) < 20: # можно увеличить количество найденных R для уверенности
        binary = bin(N)[2:] # двоичная запись без '0b'
        if N % 2 == 0: # чётное: слева и справа по две единицы
            new_binary = '11' + binary + '11'
        else: # нечётное: слева единица, справа два нуля
            new_binary = '1' + binary + '00'
        R = int(new_binary, 2)
        if R > 119:
            found.append(R)
        N += 1
    # Минимальное из найденных
    print(min(found))
if __name__ == "__main__":
    solve()
```

Ответ: 123

Решение (Вариант 2)

При выполнении этого задания можно использовать вычисления в ЭТ:

	A	B	C	D	E	F	G	H
1	N	Четность N	binary_N	new_binary_N	R	R>119	Ответ:	123
2	1	1	1	1100	12			
3	2	0	10	111011	59			
4	3	1	11	11100	28			
5	4	0	100	1110011	115			
6	5	1	101	110100	52			
7	6	0	110	1111011	123	123		
8	7	1	111	111100	60			
9	8	0	1000	11100011	227	227		
10	9	1	1001	1100100	100			
11	10	0	1010	11101011	235	235		
12	11	1	1011	1101100	108			
13	12	0	1100	11110011	243	243		
14	13	1	1101	1110100	116			
15	14	0	1110	11111011	251	251		
16	15	1	1111	1111100	124	124		
17	16	0	10000	111000011	451	451		
18	17	1	10001	11000100	196	196		
19	18	0	10010	111001011	459	459		
20	19	1	10011	11001100	204	204		
21	20	0	10100	111010011	467	467		
22	21	1	10101	11010100	212	212		
23	22	0	10110	111011011	475	475		
24	23	1	10111	11011100	220	220		
25	24	0	11000	111100011	483	483		

Столбец A заполним значениями N (чтобы не анализировать возможный диапазон таких чисел, можно взять для уверенности от 1 с запасом до 20, 30, ... 100 и т.д.)

Столбец B: определяем четность N по формуле: =ОСТАТ(A2;2)

Столбец C: выполняем п. а) заданного алгоритма по формуле: =ОСНОВАНИЕ(A2;2)

Столбец D: выполняем п. б) заданного алгоритма по формуле: =ЕСЛИ(B2=0;"11"&C2&"11";"1"&C2&"00")

Столбец E: переводим полученный в столбце D результат в десятичную систему счисления по формуле: =ДЕС(D2;2)

Столбец F: отделяем из полученных в столбце E чисел те, которые превышают 119 с помощью формулы: =ЕСЛИ(E2>119;E2;»»»)

Ответ находим, используя функцию =МИН(F:F).

Ответ: 123

Анализ ошибок

Наиболее быстрый способ выполнения этого задания, это применение навыков программирования. Это возможно лишь при владении языками и методами программирования. В противном случае, выполнение задания занимает гораздо больше времени.

Неверные ответы участников практически не повторяются, что говорит о случайном характере ошибок. Чаще, это неверное прочтение задания (ищут не минимальный, а максимальный результат, или выводят результат в двоичной форме, а не в десятичной системе счисления, или исходное число, из которого получен нужный результат).

Способы избежать таких ошибок:

Если не выйти на проверку заданных условий, то неважно, какой инструмент решения задачи будет выбран, так как в случае правильной проверки необходимых условий задание легко выполняется разными способами при хорошем владении методами рассуждения или представленным инструментарием в виде доступного ПО.

При подготовке обучающихся важно сформировать комплекс умений: читать и выполнять алгоритм, переводить числа в разные системы счисления; отделять цифры числа, логически рассуждать. Здесь важную роль играет уровень сформированности метапредметных результатов п. 1.3. кодификатора ЕГЭ по информатике.

В таблице 2-14 видим, что 2/3 обучающихся из группы приступали к выполнению этого задания, то есть в целом тема им знакома и они были уверены в правильном решении, но допускали ошибки, значит, необходимо проводить целенаправленную работу над этими ошибками при подготовке в школе, использовать разные способы перевода чисел в разные системы счисления, учиться работать с ЭТ, тем более эти навыки требуются и в других заданиях: 3, 9 и пр.

Несформированные метапредметные умения	Проявление в типичных ошибках
Базовые логические действия (1.1.3, 1.1.4.)	Неумение выявить закономерности в преобразовании чисел.

Несформированные метапредметные умения	Проявление в типичных ошибках
	Неверное определение условий (чётное/нечётное). Отсутствие системности в переборе вариантов.
Базовые исследовательские действия (1.2.4, 1.2.5, 1.2.7)	Неумение построить модель вычислений. Отсутствие проверки промежуточных результатов. Неверно выделены шаги и план решения.

Задание 6

Тема: Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов.

Уровень сложности: базовый.

Рекомендуемое время выполнения: 4 минуты.

Проверяется умение определять возможные результаты работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов.

Проверяемые элементы содержания: Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат.

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Умение анализировать алгоритмы с использованием таблиц трассировки; определять без использования компьютера результаты выполнения несложных программ, включающих циклы, ветвления и подпрограммы, при заданных исходных данных.

Проверяемые метапредметные результаты: Базовые логические действия. Базовые исследовательские действия.

Что нужно знать:

выполнять ручную прокрутку программы для исполнителя, в которой используется цикл;
строить на координатной плоскости фигуру, которую нарисует Черепаха (при ее известном начальном положении);
анализировать чертеж и строить математическую модель вычисления количества целочисленных точек, площади и пр.

Пример формулировки задания (используется открытый вариант №307)

Исполнитель Черепаха действует на плоскости с декартовой системой координат. В начальный момент Черепаха

находится в начале координат, её голова направлена вдоль положительного направления оси ординат, хвост опущен. При опущенном хвосте Черепаха оставляет на поле след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения.

У исполнителя существует 6 команд: Поднять хвост, означающая переход к перемещению без рисования; Опустить хвост, означающая переход в режим рисования; Вперёд n (где n – целое число), вызывающая передвижение Черепахи на n единиц в том направлении, куда указывает её голова; Назад n (где n – целое число), вызывающая передвижение в противоположном голове направлении; Направо m (где m – целое число), вызывающая изменение направления движения на m градусов по часовой стрелке; Налево m (где m – целое число), вызывающая изменение направления движения на m градусов против часовой стрелки.

Запись Повтори k [Команда1 Команда2 ... Команда S] означает, что последовательность из S команд повторится k раз.

Черепахе был дан для исполнения следующий алгоритм:

Повтори 3 [Вперёд 27 Направо 90 Вперёд 12 Направо 90]

Поднять хвост

Вперёд 4

Направо 90

Вперёд 6

Налево 90

Опустить хвост

Повтори 3 [Вперёд 83 Направо 90 Вперёд 77 Направо 90]

Определите, сколько точек с целочисленными координатами находятся внутри области пересечения фигур, ограниченных заданными алгоритмом линиями, включая точки на границах этого пересечения.

Решение (Вариант 1)

Разберемся с задачей и поэтапно выполним предложенный алгоритм, определив вид полученного чертежа. Это можно сделать аналитически и получить схематический чертеж на черновике или в графическом редакторе, а можно использовать исполнителя «Черепаха» в среде КуМир. Преобразованный алгоритм для КуМир выглядит так:

использовать Черепаха

алг

нач

. опустить хвост

. нц 3 раз

. . вперед (27)

. . вправо (90)

. . вперед (12)

. . вправо (90)

. кц

. поднять хвост

. вперед (4)

. вправо (90)

. вперед (6)

. влево (90)

. опустить хвост

. нц 3 раз

. . вперед (83)

. . вправо (90)

. . вперед (77)

. . вправо (90)

. кц

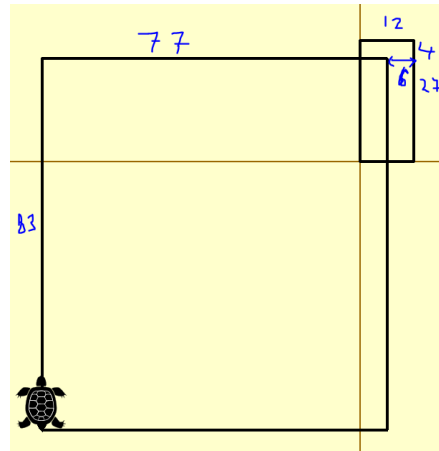
кон

Результаты анализа или выполнения алгоритма для Черепахи:

Мы получили два замкнутых контура: первый прямоугольник размером 27×12 , второй прямоугольник размером 83×77 .

Эти контуры перекрываются частично.

Примерное изображение чертежа:



Необходимо посчитать количество целых точек внутри пересечения области двух фигур, включая границы.

Здесь важно понимать порядок выполнения операций и увидеть, что за 2 итерации первого цикла прямоугольник замкнут, на третьей итерации Черепаха идет по границе прямоугольника повторно. Важно увидеть отступы в 4 и 6 единичных отрезков, которые Черепаха делает с поднятым хвостом. При работе в среде КуМир – это легко понять, если выполнять алгоритм по шагам. При аналитическом решении нужно по шагам повторить алгоритм на схематическом чертеже и отметить все отступы.

Получаем, что пересечение фигур – это прямоугольник 6 x 23.

Учтем, что если длина отрезка 6 единиц, то количество точек с целочисленными координатами на нем $6+1=7$.

Тогда количество целых точек внутри пересечения области двух фигур, включая границы равно $7 \cdot 24 = 168$

Ответ: 168

Решение (Вариант 2)

Этот вариант для программирующих обучающихся.

Пример программы на языке Python:

```
def solve():
```

```
    # Первая фигура: Повтори 3 [Вперёд 27 Направо 90 Вперёд 12 Направо 90]
```

```
    # Начало: (0, 0), голова вверх (ось Y)
```

```
x, y = 0, 0
angle = 90 # градусов от оси X (вверх)
```

```
def forward(dist):
    nonlocal x, y, angle
    import math
    rad = math.radians(angle)
    x += dist * math.cos(rad)
    y += dist * math.sin(rad)
```

```
# Рисуем первый прямоугольник
points1 = [(0, 0)]
for _ in range(3):
    forward(27)
    points1.append((round(x), round(y)))
    angle -= 90 # Направо 90
    forward(12)
    points1.append((round(x), round(y)))
    angle -= 90
```

```
# Определяем границы первого прямоугольника
xs1 = [p[0] for p in points1]
ys1 = [p[1] for p in points1]
x1_min, x1_max = min(xs1), max(xs1)
y1_min, y1_max = min(ys1), max(ys1)
# Вторая фигура:
# Поднять хвост, Вперёд 4, Направо 90, Вперёд 6, Налево 90, Опустить хвост
# Это смещает начальную точку
x, y = 0, 0
```

```

angle = 90
forward(4)    # вверх на 4
angle -= 90   # Направо 90 → вправо (0°)
forward(6)    # вправо на 6
angle += 90   # Налево 90 → снова вверх (90°)

```

```

start_x, start_y = round(x), round(y)

```

```

# Теперь рисуем второй прямоугольник

```

```

points2 = [(start_x, start_y)]
for _ in range(3):
    forward(83)
    points2.append((round(x), round(y)))
    angle -= 90
    forward(77)
    points2.append((round(x), round(y)))
    angle -= 90

```

```

xs2 = [p[0] for p in points2]
ys2 = [p[1] for p in points2]
x2_min, x2_max = min(xs2), max(xs2)
y2_min, y2_max = min(ys2), max(ys2)

```

```

# Пересечение прямоугольников

```

```

x_min = max(x1_min, x2_min)
x_max = min(x1_max, x2_max)
y_min = max(y1_min, y2_min)
y_max = min(y1_max, y2_max)

```

```

if x_min > x_max or y_min > y_max:
    print(0)
    return

```

```

# Количество целочисленных точек внутри и на границе
# Для прямоугольника от x_min до x_max включительно
# и от y_min до y_max включительно
count = (x_max - x_min + 1) * (y_max - y_min + 1)
print(count)

```

```

if __name__ == "__main__":
    solve()

```

Ответ: 168

Анализ ошибок

Эта задача вызывает трудности у школьников чаще всего из-за ряда распространенных ошибок. Вот наиболее часто встречающиеся проблемы и советы по их устранению:

1. Неправильно определены координаты фигур:

Некоторые обучающийся неправильно определяют верхние/нижние, левые/правые границы фигур. Например, считают, что прямоугольник расположен иначе, чем он фактически размещен.

Решение: Всегда внимательно следуйте инструкциям, отмечайте конечные точки, обращая внимание на исходное положение черепахи, повороты черепахи и направление движений.

2. Ошибочный расчет пересечения:

Многие ошибаются в вычислении точек на пересечении. Часто проблема возникает из-за неправильного понимания границ пересечения или отсутствия учета частичных клеток.

Решение: Используйте аккуратный геометрический подход, выделяйте чётко левую верхнюю и правую нижнюю точки пересечения и применяйте правильную формулу для подсчета целочисленных точек. Не забывайте что если длина отрезка 1, то концов у него 2 и целочисленных точек будет всегда на 1 больше, если включать границы фигуры.

3. Недопонимание условий задачи или невнимательное прочтение:

Часто обучающийся начинают считать точки сразу, не разобравшись детально в порядке прохождения черепахи и формировании фигур. Часто вместо пересечения ищут объединение фигур.

Решение: Обязательно необходимо внимательно прочесть текст условия до конца, даже если на первый взгляд условие уже видели и решали такую задачу. Прежде чем приступать к решению, обязательно сделайте набросок маршрута черепахи карандашом на бумаге. Проверьте каждую команду и убедитесь, что правильно поняли траекторию движения. Это важно для моделирования алгоритма вычислений, причем важно при любом выбранном способе решения.

4. Ошибка в анализе направления движения:

Важно учитывать повороты черепахи (налево-направо) и влияние на последующие передвижения вперёд и назад.

Решение: Постоянно контролируйте направление черепахи после каждого поворота, особенно при движении назад.

5. Сложности с отсечением дробных частей:

Иногда возникают сложности с округлением и определением количества целочисленных точек на границах фигур.

Решение: Пользуйтесь функциями округления (ceil и floor) для точности расчетов в случае выбора программного способа решения. Округляйте минимальное значение вверх, а максимальное вниз, чтобы включить все необходимые клетки.

Способы избежать таких ошибок:

Если не выйти на проверку заданных условий, то неважно, какой инструмент решения задачи будет выбран, так как в случае правильной проверки необходимых условий задание легко выполняется разными способами при хорошем владении методами рассуждения или представленным инструментарием в виде доступного ПО.

При подготовке обучающихся важно сформировать комплекс умений: читать и выполнять алгоритм, логически рассуждать, моделировать, находить объединение, пересечение, площадь, периметр фигур.

В таблице 2-14 видим, что лишь 1/5 доля всех участников в группе с баллами ниже минимального не приступили к заданию. Значит тема знакома даже слабо подготовленным обучающимся и задание кажется доступным, но выполнили задание верно лишь 8,26%. Таким образом, выявлена общая низкая способность экзаменуемых моделировать и анализировать несложные алгоритмы с циклами, условными операторами, связанные с построением чертежей. Значит,

необходимо проводить целенаправленную работу над вышеперечисленными проблемами при подготовке в школе, использовать разные способы решения. Причем, эти проблемы не должны выявляться и исправляться в 10-11 классе, целенаправленное формирование необходимых умений должно начинаться с 7 класса.

В 7 классе в разделе «Компьютерная графика» при изучении темы «Графический редактор» стоит использовать задания на построение чертежей с выделением областей пересечения, объединения фигур, заданных текстовым описанием с указанием координат вершин, длин сторон и пр.

В 8 классе в разделе «Алгоритмы и программирование» при изучении тем «Исполнители и алгоритмы», «Анализ алгоритмов» стоит использовать задания не только на разработку алгоритмов для построения чертежей, но и на анализ заданного алгоритма с выделением областей пересечения, объединения фигур и пр.

В 9 классе в теме «Моделирование как метод познания» и в разделе «Алгоритмы и программирование» так же стоит применять задачи на моделирование для расчета количества точек в областях пересечения, объединения фигур и пр.

В 10-11 классах при изучении раздела «Алгоритмизация и программирование» следует включать в программу отработку тех же навыков, но с применением языка программирования.

Таким образом выполняется преемственность программ обучения и формируются умения, проверяемые на ГИА.

Несформированные метапредметные умения	Проявление в типичных ошибках
Базовые логические действия (1.1.3., 1.1.4.)	Неумение выявлять закономерности в движении исполнителя. Неверное определение границ фигур. Путаница пересечения и объединения.
Базовые исследовательские действия (1.2.4, 1.2.5, 1.2.7)	Неумение построить модель движения. Отсутствие навыка визуализации (эскиз чертежа). Неверный расчёт целочисленных точек на границах.

Задание 8

Тема: Кодирование данных, комбинаторика, системы счисления.

Уровень сложности: базовый.

Рекомендуемое время выполнения: 4 минуты.

Проверяется умение применять знания основных понятий и методов, используемых при измерении количества информации.

Проверяемые элементы содержания: Теоретические подходы к оценке количества информации. Единицы измерения количества информации. Алфавитный подход к оценке количества информации. Закон аддитивности информации. Формула Хартли. Информация и вероятность. Формула Шеннона.

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Понимание основных принципов дискретизации различных видов информации.

Проверяемые метапредметные результаты: Базовые логические действия (Устанавливать существенный признак или основания для сравнения, классификации и обобщения). Базовые исследовательские действия (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем).

Что нужно знать:

в русском языке 33 буквы: 10 гласных букв (а, у, о, ы, и, э, я, ю, ё, е), 21 согласная буква (б, в, г, д, ж, з, й, к, л, м, н, п, р, с, т, ф, х, ц, ч, ш, щ) и два знака (ь, ъ);

алфавит английского языка по написанию совпадает с латинским алфавитом и состоит из 26 букв;

принципы работы с числами, записанными в позиционных системах счисления;

если слово состоит из L букв, причем есть n_1 вариантов выбора первой буквы, n_2 вариантов выбора второй буквы и т.д., то число возможных слов вычисляется как произведение

$$N = n_1 \cdot n_2 \cdot \dots \cdot n_L$$

если слово состоит из L букв, причем каждая буква может быть выбрана n способами, то число возможных слов вычисляется как $N = n^L$

если в программе L вложенных циклов и внешний цикл выполняется n_1 раз, следующий (вложенный) n_2 раз и т.д., то команды самого внутреннего цикла будут выполняться N раз, где $N = n_1 \cdot n_2 \cdot \dots \cdot n_L$. Если $n_1 = n_2 = \dots = n_L = n$, то $N = n^L$. при увеличении n или L значение N сильно возрастает, что приводит к существенному увеличению времени выполнения программы.

при решении с помощью программы на языке Python удобно использовать функции из модуля `itertools`:

`combinations` – комбинации, например,

```
from itertools import combinations
```

```
cmb = list(combinations('ABC', 2))
```

```
print( cmb )
```

```
// Результат: [('A', 'B'), ('A', 'C'), ('B', 'C')]
```

```

permutations – перестановки, например,
from itertools import permutations
cmb = list(permutations('ABC'))
print( cmb )
// Результат: [('A', 'B', 'C'), ('A', 'C', 'B'),
// ('B', 'A', 'C'), ('B', 'C', 'A'), ('C', 'A', 'B'),
// ('C', 'B', 'A')]

```

product – декартово произведение (все возможные слова заданной длины, составленные из данного алфавита), например:

```

from itertools import product
cmb = list(product('ABC',repeat=2))
print( cmb )
// Результат: [('A', 'A'), ('A', 'B'), ('A', 'C'), ('B', 'A'),
// ('B', 'B'), ('B', 'C'), ('C', 'A'), ('C', 'B'), ('C', 'C')]

```

Пример формулировки задания (используется открытый вариант №307)

Все пятибуквенные слова, составленные из букв А, К, Ц, Е, Н, Т, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. ААААА
2. ААААЕ
3. ААААК
4. ААААН
5. ААААТ
6. ААААЦ

.....

Определите, под каким номером в этом списке стоит последнее слово с чётным номером, которое не начинается с букв Н,

Т или Ц и при этом содержит в своей записи не менее одной буквы А.

Примечание. Слово – последовательность идущих подряд букв, не обязательно осмысленная.

Решение (Вариант 1) Используем методы комбинаторики, разные системы счисления и логическое рассуждение.

Шаг 1. Правильный алфавитный порядок

В условии буквы: А, К, Ц, Е, Н, Т – но порядок дан в самом начале списка:

1. ААААА
2. ААААЕ
3. ААААК
4. ААААН
5. ААААТ
6. ААААЦ

Значит, алфавитный порядок (по возрастанию):

$A < E < K < H < T < C$

Присвоим индексы для удобства (0-5):

- $A = 0$
- $E = 1$
- $K = 2$
- $H = 3$
- $T = 4$
- $C = 5$

Это шестеричная система счисления, где каждая позиция – это разряд (от младшего к старшему, т.е. справа налево).

Нумерация в списке начинается с 1 для слова ААААА.

Шаг 2. Формула номера слова

Для слова с буквами $b_1b_2b_3b_4b_5$ (где b_1 – первая буква, b_5 – последняя):

$$\text{Номер} = (b_1 \cdot 6^4 + b_2 \cdot 6^3 + b_3 \cdot 6^2 + b_4 \cdot 6^1 + b_5 \cdot 6^0) + 1$$

где каждая буква заменяется на её индекс ($A=0$, $E=1$, $K=2$, $H=3$, $T=4$, $C=5$).

Шаг 3. Условия задачи

Нам нужно последнее слово с чётным номером, которое:

1. Не начинается с Н, Т или Ц $\rightarrow$ первая буква $b_1 \in \{А, Е, К\}$ (индексы 0, 1, 2).
2. Содержит хотя бы одну букву А (индекс 0) в любой позиции.
3. Номер чётный.

Шаг 4. Поиск максимального номера

Поскольку слова упорядочены лексикографически, самые большие номера – у слов с максимальной первой буквой среди допустимых.

Допустимые первые буквы: А (0), Е (1), К (2). Максимальная из них – К (индекс 2).

Значит, искомое слово обязательно начинается с К, иначе номер будет меньше.

Шаг 5. Диапазон номеров для слов на К

Все слова, начинающиеся с К:

Первая буква фиксирована: индекс 2.

Остальные 4 буквы – любые из 6 (от АААА до ЦЦЦЦ).

Количество таких слов: $6^4 = 1296$.

Нумерация:

Первое слово на К: КАААА, тогда индексы: 2,0,0,0,0

Номер $= (2 \cdot 6^4 + 0 + 0 + 0 + 0) + 1 = 2 \cdot 1296 + 1 = 2593$

Последнее слово на К: КЦЦЦЦ, индексы: 2,5,5,5,5

Номер $= (2 \cdot 1296 + 5 \cdot 216 + 5 \cdot 36 + 5 \cdot 6 + 5) + 1 = 2592 + 1080 + 180 + 30 + 5 + 1 = 3888$

Итак, слова на К занимают номера с 2593 по 3888.

Шаг 6. Ищем последнее чётное слово с буквой А среди слов на К

Нам нужно максимальное число N в отрезке $[2593, 3888]$, такое что:

N чётное,

в соответствующем слове (начинающемся с К) есть хотя бы одна буква А (индекс 0).

Поскольку мы ищем последнее (наибольшее N), начнём с 3888 и будем двигаться вниз с шагом 2 (только чётные), пока не встретим слово с буквой А.

Шаг 7. Перебор от 3888 вниз

Формула для слова с первой буквой К (индекс 2):

$$N = 2592 + (a \cdot 216 + b \cdot 36 + c \cdot 6 + d) + 1 = 2593 + (a \cdot 216 + b \cdot 36 + c \cdot 6 + d)$$

где a, b, c, d – индексы второй, третьей, четвёртой и пятой букв соответственно (от 0 до 5).

Обозначим $X = a \cdot 216 + b \cdot 36 + c \cdot 6 + d$. Тогда $N = 2593 + X$, где X меняется от 0 до 1295.

Чётность N : N чётно $\Leftrightarrow 2593 + X$ чётно.

2593 – нечётное, значит N чётно $\Leftrightarrow X$ нечётно.

Таким образом, нам нужно максимальное нечётное X в диапазоне $[0, 1295]$, такое что в слове $Kabcd$ есть хотя бы одна буква А (индекс 0), то есть хотя бы одно из a, b, c, d равно 0.

Шаг 8. Поиск максимального нечётного X с условием

Нам нужно максимальное нечётное X , такое что среди a, b, c, d есть хотя бы один 0.

Идём от 1295 вниз с шагом 2:

- 1295: 5,5,5,5 $\rightarrow$ нет 0
- 1293: 5,5,5,3 $\rightarrow$ нет 0
- 1291: 5,5,5,1 $\rightarrow$ нет 0
- 1289: 5,5,4,5 $\rightarrow$ нет 0
- 1287: 5,5,4,3 $\rightarrow$ нет 0

и т.д. подбираем число, проверяя разложение в пятеричной системе счисления.

Подходит число $X = 1265$ – нечётное.

Номер $N = 2593 + 1265 = 3858$.

Шаг 9. Проверка

$N = 3858$ – чётное.

Первая буква: К – подходит (не Н, Т, Ц).

Слово: К Ц Ц А Ц – содержит А.

Номер 3858 больше, чем все предыдущие найденные (которые не подходили).

Ответ: 3858

Как видно, решение, основанное на логическом рассуждении и подборе, довольно трудоемкое. Гораздо быстрее построить алгоритм и выполнить задание, написав программу.

Решение (Вариант 2)

Этот вариант для программирующих обучающихся.

Пример программы на языке Python, включающей построение отсортированного по алфавиту списка заданных слов и выбора из него слова, удовлетворяющего условиям:

Буквы в алфавитном порядке (как в условии)

```
letters = ['A', 'E', 'K', 'H', 'T', 'Ц']
```

Генерируем все слова и записываем их в список

```
words = []
```

```
for a in letters:
```

```
    for b in letters:
```

```
        for c in letters:
```

```
            for d in letters:
```

```
                for e in letters:
```

```
                    words.append(a + b + c + d + e)
```

Ищем подходящее слово, перебирая все слова из списка и нумеруя их с 1

```
result = None
```

```
for i, word in enumerate(words, start=1):
```

```
    # Проверяем условия:
```

```
    # 1. Номер чётный
```

2. Не начинается с Н, Т или Ц

3. Содержит хотя бы одну букву А

if i % 2 == 0 and word[0] not in ['Н', 'Т', 'Ц'] and 'А' in word:

result = (i, word) # запоминаем последнее подходящее слово и его номер

Выводим результат

print(f'Номер: {result[0]}, Слово: {result[1]}')

Результат: Номер: 3858, Слово: КЦЦАЦ

Ответ: 3858

Можно использовать в программе модуль `itertools` для формирования множества нужных комбинаций букв (т.е. слов). Как видно, этот способ привлекательнее, но требует умения программировать, что для школьников с базовым уровнем подготовки часто недоступно.

Анализ ошибок

В задаче подобного формата школьники нередко допускают типичные ошибки, которые приводят к неверному ответу. Вот наиболее распространённые ошибки и рекомендации, как их избежать:

1. Неправильное понимание порядка слов:

Ошибка: Многие обучающийся предполагают, что слова идут в алфавитном порядке только по первой букве, игнорируя внутренние различия или ориентируются на заданный список букв в начале условия, а не на их алфавитный порядок.

Совет: Обязательно осознайте, что порядок слов зависит от полного состава букв, и даже небольшая разница в составе букв меняет позицию слова существенно, важно ориентироваться на порядок изменения букв в заданном списке слов.

2. Недооценивание влияния повтора букв:

Ошибка: Обучающиеся часто пренебрегают возможностью повторения букв, думая, что каждая буква уникальна.

Совет: Внимательно прочитайте условие задачи: буквы могут повторяться неограниченно много раз, что значительно увеличивает количество возможных комбинаций.

3. Игнорирование ограничения на количество букв «А»:

Ошибка: Иногда школьники полагают, что требуется ровно одна буква «А» или вовсе упускают требование наличия буквы «А».

Совет: Читайте условия задачи внимательно.

4. Неверный подсчёт позиций:

Ошибка: Некоторые обучающийся пытаются рассчитать позицию слова вручную, основываясь на интуиции, что неизбежно ведёт к ошибкам.

Совет: Лучше разработать стратегию: сначала оценить количество слов, исключаемых по определенным признакам, затем просчитать точное положение интересующего слова.

5. Проблемы с чётностью номера:

Ошибка: Частая ошибка – забыть, что нужен чётный номер. Многие находят слово с нечётным номером и делают выводы преждевременно.

Совет: Никогда не забывайте проверять чётность найденного номера, если это указано в условии задачи, опять же, прежде нужно внимательно прочесть условие задачи.

6. Переусложнение задачи:

Ошибка: Нередко школьники усложняют задачу чрезмерно сложными формулами или вычислениями, пытаясь показать глубину знания математики. На деле при подсчете допускаются арифметические ошибки или ошибки в модели расчета.

Совет: Будьте проще. Эта задача решается базовыми методами комбинаторики и здравым смыслом. Нет нужды придумывать сложные конструкции.

Способы избежать таких ошибок:

Самая важная рекомендация – при подготовке школьников акцентировать внимание на разборе условия, внимательном его прочтении. Прежде, чем приступать к решению, нужно выделить все ограничения и требования к ответу. Большинство ошибок происходят из-за невнимания к деталям или поспешных выводов. Стоит включать в учебный материал задания, нацеленные на работу с условием: «Расставь данные буквы по алфавиту», «Продолжи данный алфавитный порядок слов до 20», «Сколько слов, начинающихся на букву «К» будет в этом списке?», «Сколько всего слов в списке?», «Под каким номером в полном списке будет стоять слово «ЦТТАК»?», «Выдели все требования к искомому слову». Такие приемы позволят учащимся сконцентрироваться на тексте задания, повторить комбинаторику

и выбрать правильный способ и алгоритм решения.

Пропедевтически стоит включать такие задания при изучении темы «Равномерные коды» в разделе «Представление информации» в 7 классе, а также, включать задания по типу задания №8 из КИМ ЕГЭ при изучении раздела «Алгоритмизация и программирование» в 9-11 классах, усложняя формулировки условий постепенно.

Несформированные метапредметные умения	Проявление в типичных ошибках
Базовые логические действия (1.1.1., 1.1.4.)	Неверное определение алфавитного порядка. Неумение установить основание системы счисления. Игнорирование условий задачи (чётность, наличие буквы).
Базовые исследовательские действия (1.2.4, 1.2.5, 1.2.7)	Отсутствие проверки промежуточных и итоговых результатов. Неверный подбор критериев для проверки условий. Неумение выбрать эффективный способ решения.

Задание 9

Тема: Встроенные функции в электронных таблицах.

Уровень сложности: базовый.

Рекомендуемое время выполнения: 6 минут.

Проверяется умение обрабатывать числовую информацию в электронных таблицах.

Проверяемые элементы содержания: Анализ данных с помощью электронных таблиц. Вычисление суммы, среднего арифметического, наибольшего (наименьшего) значения диапазона. Вычисление коэффициента корреляции двух рядов.

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Умение использовать электронные таблицы для анализа, представления и обработки данных (включая выбор оптимального решения, подбор линии тренда, решение задач прогнозирования); умение использовать табличные (реляционные) базы данных и справочные системы.

Проверяемые метапредметные результаты: Работа с информацией

Что нужно знать:

для вычисления максимального, минимального и среднего арифметического значений диапазона (например, A1:G20) используются соответственно функции

MAX(A1:G20) МАКС(A1:G20)

MIN(A1:G20) МИН(A1:G20)

AVERAGE(A1:G20) СРЗНАЧ(A1:G20)

Слева записаны английские названия, справа – русские (выбор зависит от программы и версии операционной системы).

в списке аргументов этих функций можно указывать несколько диапазонов и адресов ячеек, разделив их точкой с запятой, например:

МАКС(A1:G20;H15;K12:Y90)

МИН(A1:G20;H15;K12:Y90)

СРЗНАЧ(A1:G20;H15;K12:Y90)

все три функции игнорируют (не учитывают) пустые ячейки и ячейки, содержащие нечисловые (например, текстовые) данные;

кроме перечисленных функций могут понадобиться и облегчить выполнение заданий следующие функции: СЧЁТ или СЧЁТЕСЛИ; СУММ или СУММЕСЛИ; НАИБОЛЬШИЙ (НАИМЕНЬШИЙ); ОСТАТ; ЦЕЛ; И; ИЛИ; ЕСЛИ.

Пример формулировки задания (используется открытый вариант №307)

Откройте файл электронной таблицы, содержащей в каждой строке пять натуральных чисел. Определите наименьший номер строки таблицы, содержащей числа, для которых выполнены оба условия:

в строке все числа различны;

удвоенная сумма максимального и минимального чисел строки больше суммы оставшихся трёх её чисел. В ответе запишите только число.

Пример представления исходных данных в файле:

	A	B	C	D	E	F
1	144	144	144	144	46	
2	123	123	123	123	393	
3	126	213	213	213	68	
4	469	469	469	469	189	
5	313	313	313	313	465	
6	449	449	449	449	92	
7	185	185	185	185	176	
8	120	120	418	261	31	
9	56	56	56	56	267	
10	254	254	254	254	356	

Решение (Excel)

Подготовительные шаги:

Скачать файл с табличными данными и открыть его в Excel.

Пусть таблица находится на листе с именем «Данные», а сами данные размещены в столбцах A-E.

Алгоритм:

Создаём вспомогательные колонки:

Столбцы F-J: содержат результаты повторов каждого числа данной строки. Используем формулу $F1 = \text{СЧЁТЕСЛИ}(\$A1:\$E1;A1)$

Столбец K: проверка 1 условия с помощью формулы $K1 = \text{ЕСЛИ}(\text{СЧЁТЕСЛИ}(F1:J1;1)=5;1;0)$

Столбец L: находим сумму максимального и минимального значения из данных чисел в данной строке. Используем формулу $L1 = \text{МАКС}(A1:E1) + \text{МИН}(A1:E1)$

Столбец M: находим сумму остальных трех чисел в строке. Используем формулу $M1 = \text{СУММ}(A1:E1) - L1$

Столбец N: проверка 2 условия с помощью формулы $N1 = \text{ЕСЛИ}(2*L1 > M1; 1; 0)$

Скопируем по столбцам формулы в 1 строке на оставшиеся строки. Получаем расчеты:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	144	144	144	144	46	4	4	4	4	1	0	190	432	0
2	123	123	123	123	393	4	4	4	4	1	0	516	369	1
3	126	213	213	213	68	1	3	3	3	1	0	281	552	1
4	469	469	469	469	189	4	4	4	4	1	0	658	1407	0
5	313	313	313	313	465	4	4	4	4	1	0	778	939	1
6	449	449	449	449	92	4	4	4	4	1	0	541	1347	0
7	185	185	185	185	176	4	4	4	4	1	0	361	555	1
8	120	120	418	261	31	2	2	1	1	1	0	449	501	1
9	56	56	56	56	267	4	4	4	4	1	0	323	168	1
10	254	254	254	254	356	4	4	4	4	1	0	610	762	1
11	386	386	86	86	86	2	2	3	3	3	0	472	558	1
12	425	425	425	425	62	4	4	4	4	1	0	487	1275	0
13	79	464	464	464	440	1	3	3	3	1	0	543	1368	0
14	497	497	497	497	359	4	4	4	4	1	0	856	1491	1

Если с помощью фильтров отобрать строки, где оба условия выполнены (в столбцах K и N единицы), то мы увидим наименьший номер подходящей строки. Это номер 19.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	1	1	1	1								1	4	
19	44	59	221	318	66	1	1	1	1	1	1	362	346	1
22	145	268	64	283	226	1	1	1	1	1	1	347	639	1
23	496	212	77	130	293	1	1	1	1	1	1	573	635	1
24	165	184	127	391	185	1	1	1	1	1	1	518	534	1
32	375	164	19	259	237	1	1	1	1	1	1	394	660	1
36	14	423	38	337	249	1	1	1	1	1	1	437	624	1
39	391	343	310	187	459	1	1	1	1	1	1	646	1044	1
40	306	467	430	87	204	1	1	1	1	1	1	554	940	1
42	185	224	361	448	292	1	1	1	1	1	1	633	877	1
52	88	103	464	26	91	1	1	1	1	1	1	490	282	1
54	362	231	234	25	150	1	1	1	1	1	1	387	615	1
56	261	94	226	129	317	1	1	1	1	1	1	411	616	1
57	188	106	55	144	479	1	1	1	1	1	1	534	438	1
59	386	176	469	66	27	1	1	1	1	1	1	496	628	1
61	496	333	78	47	197	1	1	1	1	1	1	543	608	1

Важно понимать, что первая строка занята под маркеры фильтров, но она нам не нужна, так как для данных этой

троки условия не выполнялись. В противном случае нужно было бы брать номер этой строки и не забыть про нее.

Можно решать задачу с помощью программирования, но, как правило, для школьников с базовым уровнем подготовки этот метод недоступен.

Ответ: 19

Анализ ошибок

При решении подобной задачи обучающиеся зачастую сталкиваются с рядом типичных трудностей и совершают распространенные ошибки:

1. Невнимательность к условию задачи:

Самая распространённая ошибка связана с непониманием формулировки задачи. Например, обучающийся может пропустить фразу "удвоенная сумма", и посчитать просто сумму макс и мин. Или находить не номер максимальной строки, а количество строк с указанными числами.

Совет: читать условие задачи медленно и внимательно, делая акцент на ключевых словах («все числа различны», «удвоенная сумма максимального и минимального», «больше суммы оставшихся трёх»), выделять (желательно, выписывая) все условия заданные в формулировке.

2. Неграммотное использование вспомогательных столбцов:

Зачастую обучающийся запутывается в организации вспомогательных столбцов для промежуточных вычислений, что приводит к путанице в формулах.

Совет: организовать рабочий лист аккуратно, создав отдельные столбцы для меток повторяющихся чисел, расчёта средних значений и прочего, желательно выделять столбцы цветами.

3. Недостаточное тестирование примеров при решении с помощью программирования:

Без тестирования решения на конкретных простых примерах можно допустить ошибку, даже не подозревая о ней.

Совет: перед сдачей решения проверьте алгоритм на небольшом тестовом файле, проверив несколько строк вручную, убедившись, что все числа в них обрабатываются формулами верно.

4. Паника при работе с большими объёмами данных:

Вид большого объёма данных пугает многих обучающихся, и они теряют концентрацию, совершая глупые ошибки.

Совет: действовать спокойно и структурированно, работая маленькими частями, разбивая задачу на подзадачи: найдите первое повторяющееся число, затем вычислите среднее, и только потом принимайтесь за сравнение.

Итоги:

Большинство ошибок связано с поверхностным прочтением условий задачи, небрежностью при работе с данными и недостаточной подготовкой промежуточных этапов вычислений. Чтобы справиться с подобными задачами эффективно, старайтесь соблюдать спокойствие, планировать шаги заранее и проводить предварительное тестирование решений.

Способы избежать таких ошибок:

Итак, подытожим.

Самая важная рекомендация – при подготовке школьников акцентировать внимание на разборе условия, внимательном его прочтении, 80% правильности решения зависит от проработки и понимания формулировки. Прежде, чем приступать к решению, нужно выделить все ограничения и требования к ответу. Стоит привить у школьников привычку выполнять постановку задачи, отвечая на вопросы «Что дано?», «Что нужно найти?», «Каковы условия поиска? Сколько их?», выписывая это на бумагу.

Далее нужно приучить школьников разбивать задачу на подзадачи, это поможет выделить этапы построения алгоритма.

На каждом этапе решения необходимо моделирование: чаще ошибки здесь возникают при составлении формул: незнание функций, неверный порядок операций, пропуск отдельных условий, неверное составление формулы для проверки конкретного условия... Именно этап моделирования (составления формул для вычисления) является непреодолимой преградой для школьника. При анализе статистических результатов мы видели, что задание 3 (базового уровня) и даже задание 18 (повышенного уровня) экзаменуемые выполняют лучше, чем задание 9. То есть школьники умеют составлять и применять в электронных таблицах основные встроенные функции, но составить и последовательно применить каскад формул для проверки данных на выполнение конкретных заданных условий – не могут. А именно эти этапы: формализация, моделирование, разбиение задачи на подзадачи очень важны в алгоритмизации и программировании. По сути, при выполнении таких заданий, как задание 9, школьники учатся алгоритмизации и программированию, используя язык ЭТ!

Процент выполнения задания крайне низок во всех группах участников ЕГЭ, но задание проверяет базовый уровень подготовки к работе с табличными данными. Больше внимания работе с электронными таблицами уделяется в рабочей программе углубленного курса обучения информатике в школе, однако, эта тема представлена и в программе

базового курса обучения информатике как в основной, так и в старшей школе. Поэтому учить составлять формулы по выделенным в задании условиям необходимо начинать в 9 классе в разделе «Моделирование как метод познания» (название из федеральной рабочей программы) и вернуться к отработке перечисленных выше навыков в теме в 11 классе в теме «Анализ данных с помощью электронных таблиц».

Важно применять задания на работу с формулировками, условиями, учить разбивать задачи на подзадачи, моделировать, тестировать промежуточные и итоговый результат. Это важные начальные этапы при разработке алгоритмов решения задач на ЭВМ любыми средствами.

Несформированные метапредметные умения	Проявление в типичных ошибках
Работа с информацией (1.3.2., 1.3.4.)	Неумение разбивать задачу на подзадачи. Отсутствие навыка формализации и моделирования. Неверное выделение данных для анализа.
Базовые исследовательские действия (1.2.4, 1.2.5)	Отсутствие навыка проверки рассчитанных значений. Неумение последовательно применять каскад формул.

○ *Предположение о достигнутых и недостигнутых метапредметных результатах ФГОС*

Можно предположить, что частично достигнуты такие метапредметные результаты ФГОС как «Работа с информацией»:

владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления (таблицы и графы).

использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач (при работе с текстовыми редакторами).

Не достигнуты в рассматриваемой группе прежде всего метапредметные результаты: *базовые логические действия, базовые исследовательские действия.*

Рассмотрим в разрезе заданий с наиболее низким результатом.

Задание 5 и задание 6

Кроме описанных выше ошибок при выполнении этих заданий, демонстрирующих слабые предметные результаты обучения по выделенным содержательным элементам, на успешность их выполнения могла повлиять прежде всего

слабая сформированность следующих метапредметных умений:

Овладение универсальными учебными познавательными действиями: базовые логические действия (1.1.3., 1.1.4.), базовые исследовательские действия (1.2.4, 1.2.5, 1.2.7).

Примеры ошибок по этой причине: неудачно выбранный способ рассуждения, неправильно построенная модель, неверный анализ информации – многие экзаменуемые не увидели части условий; отсутствие проверки и оценки промежуточных результатов, пропуск недостоверно полученных значений, не подходящих под условия; неверно выделены шаги и план решения.

Задание 8

На успешность выполнения могла повлиять прежде всего слабая сформированность следующих метапредметных умений:

Овладение универсальными учебными познавательными действиями: базовые логические действия (1.1.1., 1.1.4.), базовые исследовательские действия (1.2.4, 1.2.5, 1.2.7).

Примеры ошибок по этой причине: неверно выбран алфавит и упорядочивание слов; отсутствие достоверности проверки промежуточных и итоговых результатов; неверный подбор критериев для проверки условий, выделенных в формулировке задания; отсутствие навыков выполнения задания разными методами для проверки результата; неумение выбрать наиболее эффективный способ решения (комбинаторика, нумерация и применение систем счисления, программирование и пр.)

Задание 9

Низкий процент выполнения задания прежде всего связан с несформированностью навыков формализации и моделирования, относящимися к навыкам работы с информацией (1.3.2., 1.3.4.).

Примеры ошибок по этой причине: неумение разбивать задачу на подзадачи, сортировать, выделять данные ведет к путанице порядка выполнения и выделению ненужных ячеек; отсутствие навыка проверки рассчитанных значений для конкретного диапазона влечет накопление ошибок.

3.2.1.3. Рекомендации для учителей по совершенствованию преподавания учебного предмета группе обучающихся с минимальным уровнем подготовки

При работе с обучающимися с минимальным уровнем подготовки следует акцентировать внимание на следующих рекомендациях.

Рекомендация 1: Сфокусироваться на освоении «базового ядра» заданий для гарантированного преодоления порога обучающимися на экзамене.

Проблема: Обучающиеся этой группы имеют крайне низкие результаты по большинству заданий, но при этом показывают высокую успешность в заданиях №1 и №10. Анализ показывает неиспользованный потенциал в заданиях №2, №3, №4 и №7 (Таб. 2-16, стр. 35).

Рекомендация (методика «Приоритетный минимум»):

Разработайте для этой категории обучающихся индивидуальные или групповые образовательные маршруты, сфокусированные на отработке строго ограниченного набора тем, которые в сумме дадут проходной балл (6-7 первичных баллов). Освойте следующие темы в порядке приоритета:

Задание 7 (Измерение информации): Отработка формул для расчета объема графической и звуковой информации.

Задание 4 (Кодирование и декодирование): Решение задач на условие Фано. Используйте методику построения дерева кода.

Задание 2 (Логика): Решение задач на построение таблиц истинности (пошагово, без использования сложных преобразований).

Задание 3 (Базы данных): Отработка навыков поиска, фильтрации и сортировки в табличных базах данных.

Организация процесса: Выделите на уроках время для «минуток повторения» по этим темам. Используйте систему «накопительных карточек», где обучающийся отмечает решенные задачи по каждой теме, переходя к следующей только после успешного выполнения 5-7 однотипных заданий из открытого банка ФИПИ.

Рекомендация 2: Целенаправленно формировать базовые исследовательские и логические действия через моделирование и пошаговое решение задач.

Проблема: Обучающиеся демонстрируют недостаточную сформированность метапредметных умений (базовых логических и исследовательских действий), что проявляется в неумении выявлять закономерности, строить алгоритмы и анализировать результаты, особенно в заданиях №5, №6, №8, №9 (стр. 36, 46, 53, 57).

Рекомендация (приемы работы):

Задания №5, №6 («Исполнители и алгоритмы»): Вместо решения сложных задач сразу, практикуйте «ручную прокрутку» (трассировку) алгоритмов на небольших наборах данных. Научите обучающихся составлять таблицы трассировки, пошагово записывая значения переменных и координаты исполнителя. Важно: сначала объясняйте на примерах с бумагой и карандашом, и только после этого показывайте решение в среде КуМир или на Python.

Задание 8 («Комбинаторика и системы счисления»): Для непрограммирующих обучающихся используйте прием

«перевод в систему счисления». Научите их воспринимать список слов как числа в системе счисления с основанием, равным количеству букв в алфавите. Отрабатывайте этот навык на простых примерах (например, слова из 3 букв), постепенно усложняя.

Задание 9 («Электронные таблицы»): Вместо решения задач целиком, научите разбивать условие на отдельные логические блоки. Например, для нахождения строки с уникальными числами: сначала отработайте формулу СЧЁТЕСЛИ, затем – формулу для проверки равенства, и только потом – объединение этих условий в сложный фильтр. Используйте метод «от простого к сложному»: сначала решите задачу, где нужно найти только сумму или только максимум, а затем постепенно добавляйте новые условия.

Рекомендация 3: Развивать навыки работы с информацией и внимание к деталям через практику «умного чтения».

Проблема: Основная масса ошибок в группе возникает из-за невнимательного прочтения условия: обучающийся ищут не то, что требуется (максимум вместо минимума, нечетный номер вместо четного, объединение фигур вместо пересечения), или неверно интерпретируют исходные данные (стр. 40, 46, 53, 57).

Рекомендация (технология «Анализ условия»):

Внедрите обязательный этап «Парсинга задачи» перед любым решением. Для этого попросите обучающихся на черновике выписать три вещи:

«Что дано?» (Исходные данные, алфавит, последовательность команд).

«Что нужно найти?» (Формат ответа: десятичное число, номер слова, количество точек и т.д.).

«Каковы все условия?» (Выпишите все ограничения из задачи в виде маркированного списка).

Пример для задания №6: Предложите обучающимся перед началом вычислений сделать эскиз (набросок) фигуры на черновике, отмечая координаты всех поворотов. Это помогает визуализировать пересечение и избежать ошибок с объединением.

Пример для задания №8: Давайте задания на составление обратной задачи: «Придумай условие к слову с номером N» или «Найди ошибку в моем решении». Это приучает к внимательной работе с условием и развивает навыки самоконтроля.

Рекомендация 4: Организовать межпредметную и пропедевтическую работу для формирования навыков алгоритмизации с 7 класса.

Проблема: Дефициты по темам (например, исполнители Черепаха, поиск в ЭТ), проверяемые на ЕГЭ, не должны формироваться только в 10-11 классе. Анализ показывает, что успешность выполнения заданий №6 и №9 крайне низка, даже несмотря на то, что они кажутся учащимся знакомыми (стр. 47, 58).

Рекомендация (системная работа):

Преемственность: Начиная с 7 класса, включайте в уроки задания на формальное исполнение алгоритмов (например, задачи на построение фигур по координатам в графическом редакторе) и на логическое моделирование (задачи на поиск закономерностей). Это заложит базу для метапредметного умения «выявлять закономерности и противоречия».

Работа с данными: В курсе информатики 9-10 класса усильте практическую часть по работе с электронными таблицами. Не ограничивайтесь простыми вычислениями, давайте задачи на условные и логические функции (ЕСЛИ, И, ИЛИ) для фильтрации данных, так как это напрямую связано с алгоритмическим мышлением.

Выбор инструмента: Обсуждайте с обучающимися, какой способ решения (программирование или ЭТ) для них проще и быстрее в каждой конкретной задаче. Предлагайте решать задания разными способами, чтобы обучающийся мог выбрать наиболее комфортный для себя. Это также является элементом формирования метапредметного умения (оценка эффективности способа решения).

3.2.2. Методический анализ выполнения заданий обучающимися с низким уровнем подготовки (в группе от минимального до 60 т.б.), включая методические рекомендации для учителей

3.2.2.1. Описание предметной подготовки участников ЕГЭ с низким уровнем подготовки, анализ типичных дефицитов в подготовке

○ *Описание достигнутых предметных результатов ФГОС: освоенных тем программы, сформированных умений и т.п. (если имеются)*

Группа участников, с низким уровнем подготовки, показала наиболее высокие результаты при выполнении заданий 1, 2, 3, 4, 7 и 10 базового уровня сложности, заданий 12 и 18 повышенного уровня сложности и почти треть группы справилась с заданием 21 высокого уровня сложности. По остальным заданиям результаты низкие.

Можно констатировать, что обучающимися этой группы на высоком уровне освоены следующие элементы содержания школьной программы по информатике и умения:

1) Задание 1: Разделы/темы школьной программы: 11 кл., п. 113.7.2. Представление результатов моделирования в виде, удобном для восприятия человеком. Графическое представление данных (схемы, таблицы, графики). Умения и навыки: умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде. Метапредметные результаты: Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления.

2) Задание 2: Разделы/темы школьной программы: 10 кл., п. 113.6.2. Алгебра логики. Логические операции. Таблицы истинности. Логические выражения. Логические тождества. Умения и навыки: Умение строить логическое выражение в дизъюнктивной и конъюнктивной нормальных формах по заданной таблице истинности; исследовать область истинности высказывания, содержащего переменные; решать несложные логические уравнения. Умение строить таблицы истинности и логические схемы. Метапредметные результаты: *Базовые логические действия* (Устанавливать существенный признак или основания для сравнения, классификации и обобщения. Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности.). *Базовые исследовательские действия* (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях.).

3) Задание 3: Разделы/темы школьной программы: 11 кл., п. 113.7.4. Табличные (реляционные) базы данных. Поле, запись. Ключ таблицы. Работа с готовой базой данных. Поиск, сортировка и фильтрация данных. Запросы на выборку данных. Многотабличные базы данных. Типы связей между таблицами. Умения и навыки: Владение основными сведениями о базах данных, их структуре, средствах создания и работы с ними. Умение поиска информации в реляционных базах данных. Метапредметные результаты: Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных

видов и форм представления.

4) Задание 4: Разделы/темы школьной программы: 10 кл., п. 113.6.2. Двоичное кодирование. Равномерные и неравномерные коды. Декодирование сообщений, записанных с помощью неравномерных кодов. Условие Фано. Построение однозначно декодируемых кодов с помощью дерева. Умения и навыки: Умение кодировать и декодировать информацию. Умение строить код, обеспечивающий наименьшую возможную среднюю длину сообщения при известной частоте символов. Метапредметные результаты: *Базовые исследовательские действия* (Владеть навыками учебно-исследовательской деятельности. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях.).

5) Задание 7: Разделы/темы школьной программы: 10 кл., п. 113.6.2. Кодирование изображений. Оценка информационного объёма графических данных при заданных разрешении и глубине кодирования цвета. Цветовые модели. Кодирование звука. Оценка информационного объёма звуковых данных при заданных частоте дискретизации и разрядности кодирования. Умения и навыки: Умение определять информационный объём текстовых, графических и звуковых данных при заданных параметрах дискретизации. Умение определять среднюю скорость передачи данных, оценивать изменение времени передачи при изменении информационного объёма данных и характеристик канала связи. Метапредметные результаты: *Базовые исследовательские действия* (Владеть навыками учебно-исследовательской деятельности. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду.).

6) Задание 10: Разделы/темы школьной программы: 10 кл., п. 113.6.3. Текстовый процессор. Средства поиска и автозамены в текстовом процессоре. Структурированные текстовые документы. Сноски, оглавление. Правила цитирования источников и оформления библиографических ссылок. Умения и навыки: владение навыками работы с операционными системами и основными видами программного обеспечения для решения учебных задач по выбранной специализации. Метапредметные результаты: Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм

представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.

7) Задание 12. Разделы/темы школьной программы: 11 кл., п. 113.7.3. Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат. Умения и навыки: Понимание базовых алгоритмов обработки числовой и текстовой информации (запись чисел в позиционной системе счисления, делимость целых чисел; нахождение всех простых чисел в заданном диапазоне; обработка многоразрядных целых чисел; анализ символьных строк и других), алгоритмов поиска и сортировки. Метапредметные результаты: *Базовые логические действия* (Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения.). *Базовые исследовательские действия* (Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях.).

8) Задание 18. Разделы/темы школьной программы: 11 кл., п. 113.7.4. Анализ данных с помощью электронных таблиц. Вычисление суммы, среднего арифметического, наибольшего (наименьшего) значения диапазона. Решение задач оптимизации с помощью электронных таблиц. Умения и навыки: Умение использовать электронные таблицы для анализа, представления и обработки данных (включая выбор оптимального решения, подбор линии тренда, решение задач прогнозирования); умение использовать табличные (реляционные) базы данных и справочные системы. Метапредметные результаты: *Работа с информацией* (Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности).

9) Разделы/темы школьной программы: 11 кл., п. 113.7.2. Дискретные игры двух игроков с полной информацией. Построение дерева перебора вариантов, описание стратегии игры в табличной форме. Выигрышные и проигрышные позиции. Выигрышные стратегии. Умения и навыки: Умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде. Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию. Метапредметные результаты: *Работа с информацией* (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных

видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.) *Коммуникативные УУД* (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств).

○ *Анализ типичных дефицитов в предметной подготовке: неосвоенные или слабо освоенные темы программы, несформированные или недостаточно сформированные умения, навыки, виды познавательной деятельности, проявляющиеся в типичных ошибках при выполнении соответствующих заданий*

Прежде всего для данной группы обучающихся стоит задача освоения предметных результатов, соответствующих базовому и повышенному уровню подготовки по информатике. Задания высокого уровня сложности не рассчитаны на данную группу обучающихся. Выделим содержательные элементы и умения, проверяемые заданиями ЕГЭ, результаты выполнения которых низкие в данной группе – это задания 5, 6, 8, 9, 19 базового уровня, 11, 12, 13, 16, 20, 22, 23.

Таблица 2-17

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП
1.	Задание 5 <u>Элементы содержания:</u> Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат. <u>Умения и навыки:</u> Умение анализировать алгоритмы с использованием таблиц трассировки; определять без использования компьютера результаты выполнения несложных программ, включающих циклы, ветвления и подпрограммы, при заданных исходных данных. <u>Метапредметные результаты:</u> <i>Базовые логические действия</i> (Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности). <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской деятельности. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в	Базовый уровень: 11 кл., п. 113.7.3. Углубленный уровень: 10 кл., п. 114.6.3

	ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду.)	
2.	Задание 6 <u>Элементы содержания:</u> Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат. <u>Умения и навыки:</u> Умение анализировать алгоритмы с использованием таблиц трассировки; определять без использования компьютера результаты выполнения несложных программ, включающих циклы, ветвления и подпрограммы, при заданных исходных данных. <u>Метапредметные результаты:</u> <i>Базовые логические действия</i> (Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности). <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской деятельности. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду.)	Базовый уровень: 11 кл., п. 113.7.3 Углубленный уровень: 10 кл., п. 114.6.3
3.	Задание 7 <u>Элементы содержания:</u> Кодирование изображений. Оценка информационного объёма графических данных при заданных разрешении и глубине кодирования цвета. Цветовые модели. Кодирование звука. Оценка информационного объёма звуковых данных при заданных частоте дискретизации и разрядности кодирования. Скорость передачи данных. <u>Умения и навыки:</u> Умение определять информационный объём текстовых, графических и звуковых данных при заданных параметрах дискретизации. Умение определять среднюю скорость передачи данных, оценивать изменение времени передачи при изменении информационного объёма данных и характеристик канала связи. <u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-	Базовый уровень: 10 кл., п. 113.6.2. Углубленный уровень: 10 кл., п. 114.6.2

	исследовательской деятельности. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду.)	
4.	Задание 8 <u>Элементы содержания:</u> Теоретические подходы к оценке количества информации. Единицы измерения количества информации. Алфавитный подход к оценке количества информации. Элементы комбинаторики <u>Умения и навыки:</u> Знание основных понятий и методов, используемых при измерении количества информации. Понимание основных принципов дискретизации различных видов информации. <u>Метапредметные результаты:</u> <i>Базовые логические действия</i> (Устанавливать существенный признак или основания для сравнения, классификации и обобщения). <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской деятельности.)	Базовый уровень: 10 кл., п. 113.6.2 Углубленный уровень: 11 кл., п. 114.7.1
5.	Задание 19 <u>Элементы содержания:</u> Дискретные игры двух игроков с полной информацией. Построение дерева перебора вариантов, описание стратегии игры в табличной форме. Выигрышные и проигрышные позиции. Выигрышные стратегии. <u>Умения и навыки:</u> Умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде. Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию. <u>Метапредметные результаты:</u> <i>Работа с информацией</i> (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.). <i>Коммуникативные УУД</i> (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств).	Базовый уровень: 11 кл., п. 113.7.2 Углубленный уровень: 11 кл., п. 114.7.1
6.	Задание 11	Базовый уровень:

	<u>Элементы содержания:</u> Теоретические подходы к оценке количества информации. Единицы измерения количества информации. Алфавитный подход к оценке количества информации. <u>Умения и навыки:</u> Умение определять информационный объём текстовых, графических и звуковых данных при заданных параметрах дискретизации. Умение определять среднюю скорость передачи данных, оценивать изменение времени передачи при изменении информационного объёма данных и характеристик канала связи. <u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов).	10 кл., п. 113.6.2. Углубленный уровень: 10 кл., п. 114.6.2; 11 кл., п. 114.7.1
7.	Задание 13 <u>Элементы содержания:</u> Сетевые протоколы. Сеть Интернет. Адресация в сети Интернет. Разделение IP-сети на подсети с помощью масок подсетей. <u>Умения и навыки:</u> Наличие представлений о базовых принципах организации и функционирования компьютерных сетей. Умение использовать маску подсети. <u>Метапредметные результаты:</u> <i>Работа с информацией</i> (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ре-	Базовый уровень: 11 кл., п. 113.7.1 Углубленный уровень: 10 кл., п. 114.6.1

	сурсосбережения, правовых и этических норм, норм информационной безопасности.) <i>Коммуникативные УУД</i> (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств). <i>Коммуникативные УУД</i> (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств).	
8.	Задание 14 <u>Элементы содержания:</u> Системы счисления. Развёрнутая запись целых и дробных чисел в позиционной системе счисления. Свойства позиционной записи числа: количество цифр в записи, признак делимости числа на основание системы счисления. Алгоритм перевода целого числа из Р-ичной системы счисления в десятичную. Алгоритм перевода целого числа из десятичной системы счисления в Р-ичную. Перевод конечной десятичной дроби в Р-ичную. Двоичная, восьмеричная и шестнадцатеричная системы счисления, связь между ними. Арифметические операции в позиционных системах счисления. <u>Умения и навыки:</u> Умение использовать при решении задач свойства позиционной записи чисел, алгоритмы построения записи числа в позиционной системе счисления с заданным основанием и построения числа по строке, содержащей запись этого числа в позиционной системе счисления с заданным основанием; умение выполнять арифметические операции в позиционных системах счисления. <u>Метапредметные результаты:</u> <i>Базовые логические действия</i> (Устанавливать существенный признак или основания для сравнения, классификации и обобщения. Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности. Развивать креативное мышление при решении жизненных проблем).	Базовый уровень: 11 кл., п. 113.6.2 Углубленный уровень: 10 кл., п. 114.6.2
9.	Задание 15 <u>Элементы содержания:</u> Алгебра логики. Понятие высказывания. Высказывательные формы (предикаты). Кванторы существования и всеобщности. Логические операции. Таблицы истинности. Логические выражения. Логические тождества. Логические операции и операции над множествами. Законы алгебры логики. Эквивалентные преобразования логических выражений. <u>Умения и навыки:</u> Умение строить логическое выражение в дизъюнктивной и конъюнктивной нормальных формах по заданной таблице истинности; исследовать область истинности высказывания, содержащего переменные; решать несложные логические уравнения. <u>Метапредметные результаты:</u> <i>Базовые логические действия</i> (Устанавливать существенный при-	Базовый уровень: 10 кл., п. 113.6.2. Углубленный уровень: 10 кл., п. 114.6.2

	знак или основания для сравнения, классификации и обобщения. Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности. Развивать креативное мышление при решении жизненных проблем). <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов	
10.	Задание 16 <u>Элементы содержания:</u> Рекурсия. Рекурсивные процедуры и функции <u>Умения и навыки:</u> Вычисление рекуррентных выражений. <u>Метапредметные результаты:</u> <i>Базовые логические действия</i> (Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. <i>Базовые исследовательские действия</i> (Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях.)	Углубленный уровень: 10 кл., п. 114.6.3; 11 кл., п. 114.7.2.
11.	Задание 17 <u>Элементы содержания:</u> Массивы и последовательности чисел. Вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения, среднего	Базовый уровень: 11 кл., п. 113.7.3. Углубленный уровень:

арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию). Линейный поиск заданного значения в массиве. Алгоритмы работы с элементами массива с однократным просмотром массива. Сортировка одномерного массива. <u>Умения и навыки:</u> Умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива; умение использовать в программах данные различных типов с учётом ограничений на диапазон их возможных значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; умение использовать средства отладки программ в среде программирования. <u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов). <i>Регулятивные УУД 3.1 Самоорганизация</i>	10 кл., п. 114.6.3; 11 кл., п. 114.7.2
---	---

	(Самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям. Самостоятельно составлять план решения проблемы с учётом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретённый опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний.) 3.2 <i>Самоконтроль</i> (Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям. Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приёмы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению).	
12.	Задание 20 <u>Элементы содержания:</u> Дискретные игры двух игроков с полной информацией. Построение дерева перебора вариантов, описание стратегии игры в табличной форме. Выигрышные и проигрышные позиции. Выигрышные стратегии. <u>Умения и навыки:</u> Умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде. Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию. <u>Метапредметные результаты:</u> <i>Работа с информацией</i> (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.). <i>Коммуникативные УУД</i> (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств).	Базовый уровень: 11 кл., п. 113.7.2 Углубленный уровень: 11 кл., п. 114.7.1
13.	Задание 22 <u>Элементы содержания:</u> Параллельные вычисления <u>Умения и навыки:</u> Понимание основных принципов устройства и функционирования современных стационарных и мобильных компьютеров; тенденций развития компьютерных технологий; владение навыками работы с операционными системами и основными видами программного	Базовый уровень: 10 кл., п. 113.6.1. Углубленный уровень: 10 кл., п. 114.6.1.

	обеспечения для решения учебных задач по выбранной специализации. Построение математических моделей для решения практических задач. Архитектура современных компьютеров. Много-процессорные системы. <u>Метапредметные результаты:</u> <i>Работа с информацией</i> (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.).	
14	Задание 23 <u>Элементы содержания:</u> Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат. <u>Умения и навыки:</u> Владение универсальным языком программирования высокого уровня (Паскаль, Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции; умение осуществлять анализ предложенной программы: определять результаты работы программы при заданных исходных данных; определять, при каких исходных данных возможно получение указанных результатов; выявлять данные, которые могут привести к ошибке в работе программы; формулировать предложения по улучшению программного кода. <u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить	Базовый уровень: 11 кл., п. 113.7.3. Углубленный уровень: 10 кл., п. 114.6.3; 11 кл., п. 114.7.2

проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов). <i>Регулятивные УУД 3.1 Самоорганизация</i> (Самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям. Самостоятельно составлять план решения проблемы с учётом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретённый опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний.) <i>3.2 Самоконтроль</i> (Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям. Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приёмы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению).	
---	--

○ *Реалистичные «зоны роста» в части предметной подготовки для уверенного получения 60-70 т.б.*

На основе анализа данных Таблицы 2-13 для группы «от минимального до 60 т.б.» и сравнения с результатами группы «от 61 до 80 т.б.», выделены 8 заданий, которые представляют собой наиболее реалистичные «зоны роста».

Эти задания:

имеют низкий процент выполнения в текущей группе (от 17% до 54%);

демонстрируют высокий процент выполнения в группе «61–80» (от 70% до 88%);

имеют чёткий, повторяющийся алгоритм решения, не требующий сложного программирования;

не включены в список уже успешно решаемых заданий (№ 1, 2, 3, 4, 7, 10, 12, 18, 21).

Реалистичные «зоны роста» для достижения 6-70 тестовых баллов. Задание 5 (Базовый уровень)

Тема: Формальное исполнение простого алгоритма, записанного на естественном языке.

Показатель	Значение
% выполнения в группе «от мин. до 60»	36,4%
% выполнения в группе «61–80»	78,85%
Разрыв	42,45 п.п.

Почему это «зона роста»: задание проверяет умение «читать» алгоритм и выполнять его по шагам. Ошибки возникают из-за невнимательности (путают чётное/нечётное, двоичную/десятичную систему) или неумения работать с системами счисления. При этом 2/3 обучающихся группы приступают к заданию, но допускают ошибки. Треть группы справляется с этим заданием, следующая группа «61–80 т.б.» вполне успешно его выполняет.

Что отрабатывать:

- перевод чисел между системами счисления ($2 \leftrightarrow 10$);
- пошаговое выполнение алгоритма с фиксацией промежуточных результатов;
- использование таблиц трассировки для контроля вычислений;
- решение задач на бит чётности и добавление разрядов.

Задание 6 (Базовый уровень)

Тема: Определение результатов работы алгоритмов управления исполнителями (Черепаша).

Показатель	Значение
% выполнения в группе «от мин. до 60»	54,41%
% выполнения в группе «61–80»	82,31%
Разрыв	27,9 п.п.

Почему это «зона роста»: задание кажется знакомым учащимся (лишь 1/5 не приступают), но правильный ответ дают только 54%. Основные ошибки: неверное определение координат фигур, путаница объединения и пересечения, неправильный учёт точек на границах. Половина группы справляется с этим заданием, следующая группа «61–80 т.б.» вполне успешно его выполняет.

Что отрабатывать:

- пошаговое моделирование движения Черепаша на бумаге (координаты и направление);
- построение эскиза фигуры перед расчётами;
- формула подсчёта целочисленных точек внутри прямоугольника, учитывая границы: $(X_{\max} - X_{\min} + 1) \times (Y_{\max} - Y_{\min} + 1)$;
- формула подсчёта целочисленных точек внутри прямоугольника, без учета границ: $(X_{\max} - X_{\min} - 1) \times (Y_{\max} - Y_{\min} - 1)$;
- различие понятий «пересечение» и «объединение» фигур и внимательность при прочтении условий.

Задание 8 (Базовый уровень)

Тема: Кодирование данных, комбинаторика, системы счисления.

Показатель	Значение
% выполнения в группе «от мин. до 60»	32,43%
% выполнения в группе «61–80»	83,46%
Разрыв	51,03 п.п.

Почему это «зона роста»: то, что треть группы справляется с этим заданием, но имеется огромный разрыв между группами, говорит о том, что задание принципиально решаемо, но требует освоения стандартного алгоритма. Основные ошибки: неверный алфавитный порядок, игнорирование повторений букв, пропуск условий (чётность, наличие буквы).

Что отрабатывать:

восприятие списка слов как чисел в системе счисления с основанием = количеству букв;

формула номера слова: $\text{Номер} = (a_1 \cdot q^4 + a_2 \cdot q^3 + a_3 \cdot q^2 + a_4 \cdot q^1 + a_5 \cdot q^0) + 1$;

проверка чётности номера через чётность суммы разрядов;

использование простого перебора (вручную или в Excel) для небольших алфавитов.

Задание 9 (Базовый уровень)

Тема: Обработка числовой информации в электронных таблицах (сложные условия).

Показатель	Значение
% выполнения в группе «от мин. до 60»	17,48%
% выполнения в группе «61–80»	69,62%
Разрыв	52,14 п.п.

Почему это «зона роста»: самое низкое выполнение среди базовых заданий, при этом задание 3 базового уровня и задание 18 повышенного уровня на работу с ЭТ выполнены успешно. Обучающиеся умеют использовать отдельные функции (МАКС, МИН, СРЗНАЧ), но не могут составить каскад формул для проверки нескольких условий. Именно навык формализации и разбиения задачи на подзадачи здесь ключевой.

Что отрабатывать:

разбиение условия на отдельные логические блоки (например: «все числа различны» → проверка повторов через СЧЁТЕСЛИ);

пошаговое добавление условий: сначала одно, потом два, потом три;
использование вспомогательных столбцов для промежуточных вычислений;
применение функций ЕСЛИ, И, ИЛИ, СЧЁТЕСЛИ, СУММЕСЛИ.

Задание 11 (Повышенный уровень)

Тема: Подсчёт информационного объёма сообщения.

Показатель	Значение
% выполнения в группе «от мин. до 60»	23,78%
% выполнения в группе «61–80»	54,81%
Разрыв	31,03 п.п.

Почему это «зона роста»: задание повышенного уровня, но формулы расчёта просты. Ошибки возникают из-за неверного определения мощности алфавита или путаницы с равномерным/неравномерным кодированием, а также при невнимательном прочтении условия задания.

Что отрабатывать:

формула Хартли: $I = \log_2 N$ для одного символа;

расчёт общего объёма: $I_{\text{общ}} = K \times \log_2 N$;

учёт вероятностного подхода (если символы неравновероятны);

решение задач на минимальное количество бит для кодирования символа;

перевод единиц измерения информации (биты $\rightarrow$ байты $\rightarrow$ Кбайты).

Задание 13 (Повышенный уровень)

Тема: Умение использовать маску подсети.

Показатель	Значение
% выполнения в группе «от мин. до 60»	35,5%
% выполнения в группе «61–80»	71,92%
Разрыв	36,42 п.п.

Почему это «зона роста»: задание имеет стандартный алгоритм решения, основанный на поразрядной конъюнкции IP-адреса и маски подсети. Ошибки возникают из-за непонимания двоичного представления чисел или неверного

определения номера компьютера в подсети. При этом потенциал решаемости высокий – разрыв между группами значительный.

Что отрабатывать:

- перевод IP-адресов и масок в двоичную систему счисления;
- поразрядная конъюнкция (логическое И) для определения адреса сети;
- определение номера компьютера (хоста) в подсети;
- расчёт количества адресов в подсети по маске.

Задание 16 (Повышенный уровень)

Тема: Вычисление рекуррентных выражений.

Показатель	Значение
% выполнения в группе «от мин. до 60»	37,66%
% выполнения в группе «61–80»	87,12%
Разрыв	49,46 п.п.

Почему это «зона роста»: колоссальный разрыв между группами. Задание решается простым последовательным вычислением значений от $F(1)$ до $F(N)$. Не требует понимания сложной рекурсии – достаточно аккуратной подстановки. Зачастую в условии нужно упростить выражение, значение которого требуется найти и выполнить простой арифметический подсчет. При каких-то условиях лучше написать программу, но код этой программы вполне типовой и алгоритм его построения легко понять. Кроме того, не обязательно использовать язык программирования, можно алгоритм вычислений выполнить в ЭТ.

Что отрабатывать:

- алгоритм вычисления: $F(1) = \dots, F(2) = \dots, \dots, F(N) = \dots$;
- использование электронных таблиц для автоматизации расчётов (формула с ссылкой на предыдущие ячейки);
- решение задач на рекуррентные последовательности (арифметические/геометрические прогрессии).

Задание 19 (Базовый уровень)

Тема: Умение анализировать алгоритм логической игры (выигрышная позиция за один ход).

Показатель	Значение
% выполнения в группе «от мин. до 60»	42,7%

% выполнения в группе «61–80» 83,46%
 Разрыв 40,76 п.п.

Почему это «зона роста»: это базовое задание, которое по логике решения проще, чем №20 и №21 (высокий уровень). Ошибки возникают из-за неверного определения проигрышной позиции для противника или пропуска вариантов ходов. Группа «61–80» решает его на 83%, что подтверждает высокую доступность задачи при целенаправленной отработке.

Что отрабатывать:

определение выигрышных и проигрышных позиций за один ход;

алгоритм поиска: ход должен перевести игру в заведомо проигрышную позицию для противника;

построение таблицы для всех возможных значений S (количество камней);

решение задач на анализ игры с двумя кучами или одной кучей.

Итоговый список элементов содержания и умений (приоритетные темы для подготовки к 60–70 т.б.)

№ задания	Проверяемые элементы содержания	Ключевые умения и навыки
5	Формальное исполнение простого алгоритма. Восстановление исходных данных по результатам работы.	Чтение и выполнение алгоритма «вручную». Перевод чисел между системами счисления ($2 \leftrightarrow 10$). Работа с битом чётности. Использование таблиц трассировки.
6	Определение результатов работы алгоритмов управления исполнителями (Черепаша).	Пошаговое моделирование движения исполнителя. Построение чертежа по тексту алгоритма. Вычисление площади пересечения и количества целочисленных точек в фигуре.
8	Кодирование данных, комбинаторика, системы счисления.	Восприятие списка слов как чисел в системе счисления с основанием, равным количеству букв. Вычисление номера слова по формуле. Учёт чётности номера и наличия заданных букв.
9	Обработка числовой информации в электронных таблицах (сложные условия).	Разбиение задачи на логические блоки. Составление каскада формул с использованием СЧЁТЕСЛИ, ЕСЛИ, И, ИЛИ. Проверка выполнения двух и более условий одновременно.
11	Подсчёт информационного объёма сообщения.	Применение алфавитного подхода. Расчёт количества информации с учётом вероятности и мощности алфавита. Формула Хартли.
13	Умение использовать маску подсети.	Перевод IP-адресов и масок в двоичную систему. Поразрядная конъюнкция (логическое И). Определение адреса сети и номера компьютера. Расчёт количества адресов в подсети.

№ задания	Проверяемые элементы содержания	Ключевые умения и навыки
16	Вычисление рекуррентных выражений.	Последовательный расчёт значений функции от 1 до N. Использование электронных таблиц для автоматизации вычислений.
19	Анализ алгоритма логической игры (выигрышная позиция за один ход).	Определение выигрышных и проигрышных позиций за 1 ход. Алгоритм поиска: перевод игры в проигрышную позицию для противника. Построение таблицы для всех возможных значений S.

ВЫВОД:

Акцентирование отработки умений, требуемых в решении указанных 8 тем в совокупности с 9 успешно выполняемыми заданиями позволит учащимся из группы «от минимального до 60 т.б.» уверенно набрать 60–70 тестовых баллов. Все задания имеют стандартные, повторяющиеся алгоритмы решения и не требуют глубоких навыков программирования, что делает их реалистичными для освоения при систематической пошаговой отработке.

Основные методические акценты:

трассировка и моделирование (№ 5, 6).

работа с системами счисления и комбинаторика (№ 8, 13).

формализация задач в электронных таблицах (№ 9).

стандартные вычислительные алгоритмы (№ 11, 16).

игровые и логические модели (№ 19).

Рекомендуемая последовательность освоения:

Базовые задания (№ 5, 6, 8, 9, 19) → затем повышенный уровень (№ 11, 13, 16). Это позволит постепенно наращивать сложность и формировать уверенность в решении у слабо подготовленных обучающихся.

3.2.2.2. Методический анализ качества освоения метапредметных результатов ФГОС участниками ЕГЭ с низким уровнем подготовки

○ *Развернутый комментарий на основе заданий / групп заданий, на успешность выполнения которых могла повлиять достаточная или недостаточная сформированность метапредметных умений, с указанием соответствующих метапредметных умений и их проявления в типичных ошибках в ответах участников экзамена на соответствующие задания*

Для анализа типичных ошибок школьников и их причин рассмотрим наиболее unsuccessfully выполненные задания в группе обучающихся с низким уровнем подготовки, разберем примеры правильных решений и сопоставим ошибки с соответствующими метапредметными компетенциями.

Наиболее unsuccessfully выполненное задание базового уровня №9 разобрано подробно в п. 3.2.1.2.

Наиболее unsuccessfully задания повышенного уровня 14, 15, 17, 23.

Задание 14

Тема: Позиционные системы счисления.

Уровень сложности: повышенный.

Рекомендуемое время выполнения: 3 минуты.

Проверяемые элементы содержания: Системы счисления. Развёрнутая запись целых и дробных чисел в позиционной системе счисления. Свойства позиционной записи числа: количество цифр в записи, признак делимости числа на основание системы счисления. Алгоритм перевода целого числа из P -ичной системы счисления в десятичную. Алгоритм перевода конечной P -ичной дроби в десятичную. Алгоритм перевода целого числа из десятичной системы счисления в P -ичную. Перевод конечной десятичной дроби в P -ичную. Двоичная, восьмеричная и шестнадцатеричная системы счисления, связь между ними. Арифметические операции в позиционных системах счисления.

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Умение использовать при решении задач свойства позиционной записи чисел, алгоритмы построения записи числа в позиционной системе счисления с заданным основанием и построения числа по строке, содержащей запись этого числа в позиционной системе счисления с заданным основанием; умение выполнять арифметические операции в позиционных системах счисления.

Проверяемые метапредметные результаты: базовые исследовательские действия, самоорганизация, самоконтроль.

Что нужно знать:

принципы кодирования чисел в позиционных системах счисления;

чтобы перевести число, скажем, 12345_N , из системы счисления с основанием N в десятичную систему, нужно умножить значение каждой цифры на N в степени, равной ее разряду:

4 3 2 1 0 ← разряды

$$1 \ 2 \ 3 \ 4 \ 5_N = 1 \cdot N^4 + 2 \cdot N^3 + 3 \cdot N^2 + 4 \cdot N^1 + 5 \cdot N^0$$

$$N^0 = 1$$

последняя цифра записи числа в системе счисления с основанием N – это остаток от деления этого числа на N ;
 две последние цифры – это остаток от деления на N^2 , и т.д.;

число 10^N записывается единица и N нулей: $10^N = \underbrace{10 \dots 0}_N$;

число $10^N - 1$ записывается как N девяток: $10^N - 1 = \underbrace{9 \dots 9}_N$;

число $10^N - 10^M = 10^M \cdot (10^{N-M} - 1)$ записывается как $N-M$ девяток, за которыми стоят M нулей: $10^N - 10^M = \underbrace{9 \dots 9}_{N-M} \underbrace{0 \dots 0}_M$;

число 2^N в двоичной системе записывается как единица и N нулей: $2^N = \underbrace{10 \dots 0}_N$;

число $2^N - 1$ в двоичной системе записывается как N единиц: $2^N - 1 = \underbrace{1 \dots 1}_N$;

число $2^N - 2^K$ при $K < N$ в двоичной системе записывается как $N-K$ единиц и K нулей: $2^N - 2^K = \underbrace{1 \dots 1}_{N-K} \underbrace{0 \dots 0}_K$;

поскольку $2^N + 2^N = 2 \cdot 2^N = 2^{N+1}$, получаем $2^N = 2^{N+1} - 2^N$, откуда следует, что $-2^N = -2^{N+1} + 2^N$;

число 3^N записывается в троичной системе как единица и N нулей: $3^N = \underbrace{10 \dots 0}_N$;

число $3^N - 1$ записывается в троичной системе как N двоек: $3^N - 1 = \underbrace{2 \dots 2}_N$;

число $3^N - 3^M = 3^M \cdot (3^{N-M} - 1)$ записывается в троичной системе как $N-M$ двоек, за которыми стоят M нулей:
 $3^N - 3^M = \underbrace{2 \dots 2}_{N-M} \underbrace{0 \dots 0}_M$;

можно сделать аналогичные выводы для любой системы счисления с основанием a :

число a^N в системе счисления с основанием a записывается как единица и N нулей: $a^N = \underbrace{10 \dots 0}_N$

число $a^N - 1$ в системе счисления с основанием a записывается как N старших цифр этой системы счисления, то есть, цифр $(a-1)$: $a^N - 1 = \underbrace{(a-1)(a-1) \dots (a-1)}_N$

число $a^N - a^M = a^M \cdot (a^{N-M} - 1)$ записывается в системе счисления с основанием a как $N-M$ старших цифр этой системы счисления, за которыми стоят M нулей: $a^N - a^M = \underbrace{(a-1) \dots (a-1)}_{N-M} \underbrace{0 \dots 0}_M$.

Пример формулировки задания (используется открытый вариант №307)

Определите сумму всех цифр с числовым значением, превышающим 9, в 36-ричной записи числа, заданного выражением:

$$5 \cdot 7776^{2013} - 4 \cdot 1296^{2015} + 3 \cdot 216^{2017} + 4 \cdot 36^{2019} - 5 \cdot 6^{2021} - 2023.$$

Решение (Python)

Наиболее оптимально выполнять задание с помощью программирования.

```
N=5*7776**2013-4*1296**2015+3*216**2017+4*36**2019-5*6**2021-2023
```

```
s=0
```

```
while N>0:
```

```
    r=N%36
```

```
    if r>9: s+=r
```

```
    N//=36
```

```
print(s)
```

Используя встроенную арифметику Python с поддержкой длинных целых чисел находим десятичное значение N. Одновременно в одном цикле:

На каждом шаге извлекаем последнюю 36-ричную цифру числа (через % 36)

Проверяем, превышает ли она 9

Если да – добавляем к сумме

Удаляем последнюю цифру (через целочисленное деление на 36)

Повторяем, пока число не станет равно 0

Выполнять задание можно и не программируя, но алгоритм решения будет тот же, а его реализация в ЭТ будет долгой и при больших степенях может быть переполнение памяти.

Анализ ошибок

Типичные ошибки в задании 14 по информатике (работа с позиционными системами счисления) основные ошибки связаны с переполнением типов данных при ручном или неправильном коде, невнимательным чтением условий

про искомые цифры, а также с путаницей в алфавитах систем счисления.

Проблемы при написании кода на Python.

Пренебрежение «длинной арифметикой»: Попытки использовать типы с фиксированной длиной из других языков. На Python числа произвольной длины поддерживаются автоматически, но ошибки в логике циклов while при ручном делении приводят к бесконечным циклам.

Неправильный порядок цифр: Функция перевода в другую систему счисления должна возвращать список или строку, где элементы идут от старших разрядов к младшим (требуется срез `[::-1]`). Путаница с порядком искажает подсчет нулей или конкретных цифр на позициях.

Ошибки с алфавитом букв: При ручном задании строки символов для систем счисления с основанием больше 10 (например, 0123456789abcdef) часто пропускают буквы или путают их порядок. Безопаснее использовать модуль `string` (`string.digits + string.ascii_lowercase`).

Ошибки в логике и условиях задач:

Путаница в вопросе: Вместо того чтобы посчитать количество конкретных цифр (например, нулей), находят общее количество цифр в числе или сумму знаков.

Неверное знание алфавита системы: Использование цифры, равной или превышающей основание системы счисления (например, появление цифры 7 в семеричной системе или неверное восприятие букв в шестнадцатеричной).

Потеря значащих нулей или неверный срез: При анализе строки результата функции `oct()` или `hex()` в Python забывают убирать префиксы (0o, 0x), что смещает индексы и портит итоговый ответ.

Несформированные метапредметные умения	Проявление в типичных ошибках
Базовые исследовательские действия (1.2.4, 1.2.5, 1.2.7)	Неумение видеть степени в числах ($7776 = 6^5$). Отсутствие навыка перевода между системами счисления. Непонимание механизма заёма в системе счисления с основанием q (заём даёт q , а не 10).
Регулятивные УУД (Самоорганизация) (3.1)	Отсутствие системного подхода к решению. Неумение составить план: привести к общему основанию → выполнить заёмы → перейти к нужной системе → найти требуемые цифры → выполнить действие.
Регулятивные УУД (Самоконтроль) (3.2)	Отсутствие проверки промежуточных результатов. Неумение проверить ответ обратным переводом или по размерности.

Способы избежать таких ошибок:

При изучении раздела важно формировать следующие навыки:

видеть степени в числах ($7776 = 6^5$, $1296 = 6^4$);

правильно переводить числа между системами счисления;

понимать, что заём в системе счисления с основанием q даёт q (в 6-ричной – 6, а не 10);

помнить главное правило: "Последний остаток – первая цифра";

придерживаться системности в решении: привести все числа к общему основанию, выполнить заёмы (схематично, с таблицей), перейти к нужной системе счисления (группировка), найти требуемые цифры, выполнить действие (сумма, количество);

при обучении хороший эффект дает визуализация: использовать схемы разрядов и цветные таблицы;

при рассмотрении приемов решения с помощью программирования важно предупреждать частые ошибки: не забывать $N \neq 36$ (бесконечный цикл), составлять правильное условие: $r > 9$ (не $r \geq 9$ и пр.), можно использовать `string.digits + string.ascii_uppercase` и пр.

Для развития внимательности:

составлять чек-лист перед решением (Какое основание системы? Что нужно найти (сумму, количество, конкретные цифры)? Какое условие отбора цифр?);

использовать правило "Двойной проверки": перепроверять каждый шаг, особенно заёмы и группировку;

научить проверять себя:

обратным переводом (из q -ричной в десятичную);

программно (написать простой код);

по размерности (сумма не может быть слишком большой).

Задание 15

Тема: Основные понятия математической логики.

Уровень сложности: повышенный.

Рекомендуемое время выполнения: 3 минуты.

Проверяемые элементы содержания: Алгебра логики. Понятие высказывания. Высказывательные формы (предикаты). Кванторы существования и всеобщности. Логические операции. Таблицы истинности. Логические выражения. Логические тождества. Логические операции и операции над множествами. Законы алгебры логики. Эквивалентные преобразования логических выражений. Логические уравнения и системы уравнений. Логические функции. Зависи-

мость количества возможных логических функций от количества аргументов. Канонические формы логических выражений.

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Умение строить логическое выражение в дизъюнктивной и конъюнктивной нормальных формах по заданной таблице истинности; исследовать область истинности высказывания, содержащего переменные; решать несложные логические уравнения.

Проверяемые метапредметные результаты: базовые логические действия, базовые исследовательские действия.

Что нужно знать:

условные обозначения логических операций

$\neg A, \bar{A}$	не A (отрицание, инверсия)
$A \wedge B, A \cdot B$	A и B (логическое умножение, конъюнкция)
$A \vee B, A + B$	A или B (логическое сложение, дизъюнкция)
$A \rightarrow B$	импликация (следование)

таблицы истинности логических операций «И», «ИЛИ», «НЕ», «импликация» (см. презентацию «Логика»);

операцию «импликация» можно выразить через «ИЛИ» и «НЕ»:

$$A \rightarrow B = \neg A \vee B \text{ или в других обозначениях } A \rightarrow B = \bar{A} + B$$

если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем – «ИЛИ», затем – «импликация», и в самом конце – эквиваленция (тождество);

операцию «импликация» ($A \rightarrow B$) в программе можно заменить на неравенство $A \leq B$; справедливость такое замены легко доказать через таблицы истинности;

в языке Python операции сравнения (такие как $\leq$, $=$ и др.) имеют приоритет выше, чем все логические операции; это надо учитывать при заменах импликации на $\leq$ или эквивалентности на $=$;

иногда полезны формулы де Моргана²:

$$\begin{aligned} \neg (A \wedge B) &= \neg A \vee \neg B & \overline{A \cdot B} &= \bar{A} + \bar{B} \\ \neg (A \vee B) &= \neg A \wedge \neg B & \overline{A + B} &= \bar{A} \cdot \bar{B} \end{aligned}$$

для упрощения выражений можно использовать формулы

$$A + A \cdot B = A \text{ (т.к. } A + A \cdot B = A \cdot 1 + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A \text{)}$$

² *Огастес (Август) де Морган* – шотландский математик и логик.

$$A + \bar{A} \cdot B = A + B \text{ (т.к. } A + \bar{A} \cdot B = (A + \bar{A}) \cdot (A + B) = 1 \cdot (A + B) = A + B \text{)}$$

некоторые свойства импликации

$$A \rightarrow (B \cdot C) = (A \rightarrow B) \cdot (A \rightarrow C)$$

$$A \rightarrow (B + C) = (A \rightarrow B) + (A \rightarrow C)$$

Связь логики и теории множеств:

пересечение множеств соответствует умножению логических величин, а объединение – логическому сложению; пустое множество $\emptyset$ – это множество, не содержащее ни одного элемента, оно играет роль нуля в теории множеств;

универсальное множество I – это множество, содержащее все возможные элементы заданного типа (например, все целые числа), оно играет роль логической единицы: для любого множества целых чисел X справедливы равенства $X + I = I$ и $X \cdot I = X$ (для простоты мы используем знаки сложения и умножения вместо знаков пересечения $\cap$ и объединения $\cup$ множеств);

дополнение $\bar{X}$ множества X – это разность между универсальным множеством I и множеством X (например, для целых чисел $\bar{X}$ – все целые числа, не входящие в X);

пусть требуется выбрать множество A так, чтобы выполнялось равенство $A + X = I$; в этом случае множество A должно включать дополнение $\bar{X}$, то есть $A \supseteq \bar{X}$ (или «по-простому» можно записать $A \geq \bar{X}$), то есть $A_{\min} = \bar{X}$;

пусть требуется выбрать множество A так, чтобы выполнялось равенство $\bar{A} + X = I$, в этом случае множество $\bar{A}$ должно включать дополнение $\bar{X}$, то есть $\bar{A} \supseteq \bar{X}$; отсюда $A \subseteq X$, то есть $A_{\max} = X$

Пример формулировки задания (используется открытый вариант №307)

Обозначим через ДЕЛ(n , m) утверждение «натуральное число n делится без остатка на натуральное число m »; пусть на числовой прямой дан отрезок $B = [65; 85]$. Для какого наибольшего натурального числа A логическое выражение

$$\text{ДЕЛ}(x, A) \vee ((x \in B) \rightarrow \neg \text{ДЕЛ}(x, 15))$$

истинно (т.е. принимает значение 1) при любом целом положительном значении переменной x ?

Решение (Python)

Наиболее оптимально выполнять задание с помощью программирования.

```
def DEL(n, m):
```

```
return n % m == 0
```

```
def check(A):
```

```
    for x in range(1, 1000):      # достаточно проверить до 1000 так как в этом случае при различных X возможны
    все варианты
```

```
    # значений True и False во вложенных высказываниях заданного выражения
```

```
        B = (65 <= x <= 85)
```

```
        expr = DEL(x, A) or (not B or not DEL(x, 15))
```

```
        if not expr:
```

```
            return False
```

```
    return True
```

```
# Ищем максимальное A
```

```
for A in range(100, 0, -1):
```

```
    if check(A):
```

```
        print(A)
```

```
        break
```

Ответ: 75

Выполнять задание можно и не программируя:

Упростим импликацию: $(x \in B) \rightarrow \neg \text{ДЕЛ}(x, 15)$ эквивалентно $\neg(x \in B) \vee \neg \text{ДЕЛ}(x, 15)$ (по закону: $A \rightarrow B = \neg A \vee B$)

Тогда всё выражение: $\text{ДЕЛ}(x, A) \vee \neg(x \in B) \vee \neg \text{ДЕЛ}(x, 15)$

Когда выражение ложно? Оно ложно только когда все три части ложны:

$\neg \text{ДЕЛ}(x, A)$ – x не делится на A

$(x \in B)$ – x принадлежит отрезку $[65; 85]$

$\text{ДЕЛ}(x, 15)$ – x делится на 15

То есть выражение ложно только для таких x , которые одновременно: находятся в $[65; 85]$, делятся на 15, не делятся на A .

Найдём числа из $[65; 85]$, кратные 15: 75 ($65 \leq 75 \leq 85$) и 75 делится на 15. Других кратных 15 в этом промежутке

нет ($60 < 65$, $90 > 85$). Значит, единственное опасное $x = 75$. Чтобы выражение было истинно при всех x , нужно, чтобы для $x = 75$ выражение было истинно.

При $x = 75$: $x \in B$ – истинно. $\text{ДЕЛ}(x, 15)$ – истинно ($75/15=5$). Значит, $\neg\text{ДЕЛ}(x, 15)$ – ложно.

Импликация $(x \in B) \rightarrow \neg\text{ДЕЛ}(x, 15) \rightarrow \text{Истина} \rightarrow \text{Ложь} = \text{Ложь}$

Остаётся только $\text{ДЕЛ}(x, A)$. Чтобы выражение было истинно, нужно чтобы $\text{ДЕЛ}(75, A)$ было истинно, то есть A должно делить 75.

Наибольшее натуральное A , которое делит 75: Делители 75: 1, 3, 5, 15, 25, 75. Наибольший – 75.

Ответ: 75

Анализ ошибок

Две самые частые ошибки при выполнении этого задания – это: 1) незнание законов логики или неверные преобразования, 2) при организации перебора x или A неверное определение интервала перебора значений (либо перебор слишком долгий, неэффективный, либо выпускаются решения, не включенные в промежуток). Рассмотрим подробнее эти и остальные типичные ошибки:

Ключевая ошибка – *начать перебор A без предварительного анализа*, из-за чего можно упустить верхнюю границу: если $A > 75$, то при $x = 75$ (единственном числе из отрезка $[65;85]$, кратном 15) выражение гарантированно ложно, значит A не может превышать 75.

Вторая распространённая ошибка – *неправильное преобразование импликации*: путают $(x \in B) \rightarrow \neg\text{ДЕЛ}(x, 15)$ с $\neg(x \in B) \wedge \neg\text{ДЕЛ}(x, 15)$, хотя на самом деле это $\neg(x \in B) \vee \neg\text{ДЕЛ}(x, 15)$; запомните, что $A \rightarrow B = \neg A \vee B$, а не конъюнкция.

Третья ошибка – *проверять все x подряд без ограничений*, хотя достаточно анализировать только x из $[65;85]$, потому что вне этого отрезка $\neg(x \in B)$ становится истинным и всё выражение автоматически истинно.

Четвёртая – *игнорировать границы отрезка*: нужно использовать нестрогие неравенства $65 \leq x \leq 85$, то есть $\text{range}(65, 86)$, а не $\text{range}(65, 85)$, иначе потеряете $x=85$.

Пятая – *не проверить, что A натуральное (≥ 1)*, и допустить $A=0$, что сделает $\text{ДЕЛ}(x, 0)$ неопределённым.

Шестая – *найдя одно подходящее A , сразу остановиться*, хотя нужно искать максимальное, поэтому перебор должен идти от 75 вниз до 1, и первое подходящее будет ответом.

Седьмая – *не тестировать критические точки*, особенно $x=75$, где выражение может нарушиться; всегда проверяйте вручную, что при найденном A выражение истинно именно для этого x .

Восьмая – *забывать, что $\text{ДЕЛ}(x, 15)$ должно быть истинно одновременно с $x \in B$, чтобы выражение стало*

ложным; если x не кратно 15, то $\neg \text{ДЕЛ}(x, 15) = \text{Истина}$, и выражение уже не может быть ложным.

Девятая – **путать порядок действий**: сначала находим все "опасные" x (где и $x \in B$, и x кратно 15), затем для каждого требуем $\text{ДЕЛ}(x, A)$, а уже потом ищем максимальное A , которое делит все эти x ; в данной задаче опасное x только одно – 75, поэтому A должно делить 75.

Десятая – **не использовать проверочный код для верификации**, полагаясь только на ручные выкладки; всегда пишите небольшую функцию, которая перебирает x от 1 до 85 и проверяет выражение для найденного A , это поможет отловить ошибки в логике.

И последнее: не забывайте, что ответ должен быть натуральным числом, и, если ни одно A не подходит (чего в этой задаче не случится), нужно вернуть None или сообщение об ошибке, а не молча выводить 0.

Несформированные метапредметные умения	Проявление в типичных ошибках
Базовые логические действия (1.1.1., 1.1.3., 1.1.4.)	Незнание законов логики или неверные преобразования. Путаница с импликацией ($A \rightarrow B = \neg A \vee B$, а не конъюнкция). Неумение выделить "опасные" точки, где выражение становится ложным.
Базовые исследовательские действия (1.2.4, 1.2.5, 1.2.7)	Неумение организовать перебор с обоснованной границей. Отсутствие навыка упрощения логического выражения перед программированием. Неумение найти критические точки (где $x \in B$ и x кратно 15 одновременно).
Регулятивные УУД (Самоконтроль) (3.2)	Отсутствие проверки на критической точке. Неумение протестировать найденное значение A вручную.

Способы избежать таких ошибок:

Чтобы избежать подобных ошибок, важно помнить два ключевых принципа: «сначала аналитика, потом код» и «всегда ищи слабое место». Вместо того чтобы сразу бросаться перебирать A , нужно находить критические точки, где выражение может стать ложным – для этого нужно упростить логику, заменив импликацию на дизъюнкцию (чётко проговаривая формулу $A \rightarrow B = \neg A \vee B$), и затем определить, при каких x обе части (принадлежность отрезку и делимость на 15) истинны одновременно. Вне отрезка выражение автоматически истинно, поэтому проверять все x подряд не нужно – это экономит время и снижает путаницу.

Очень важно научить не игнорировать границы: всегда использовать нестрогие неравенства и проверять крайние точки отдельно. Также стоит выработать привычку сначала вручную найти теоретический максимум для A (в данном случае 75, потому что при $A > 75$ число 75 не разделится на A), а уже потом организовывать перебор сверху вниз,

останавливаясь на первом подходящем значении. При этом обязательно требовать, чтобы A было натуральным, и делать проверку на делитель нуля, если используете функцию ДЕЛ.

Закрепляйте навык написания небольших проверочных функций, которые прогоняют все опасные x и выдают, какое из них нарушает выражение – это помогает отладить логику и увидеть ошибку на конкретном числе.

Самое главное – не доверять сразу первому найденному ответу, а всегда проверять его на критической точке (вручную или кодом), чтобы убедиться, что выражение действительно истинно при всех x . Такой подход – от анализа к коду, от поиска уязвимостей к перебору с обоснованной границей – превращает решение из угадывания в строгое рассуждение и значительно снижает вероятность типичных ошибок.

Задание 17

Тема: Перебор последовательности целых чисел. Проверка делимости.

Уровень сложности: повышенный.

Рекомендуемое время выполнения: 14 минут.

Умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования.

Проверяемые элементы содержания: Массивы и последовательности чисел. Вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения, среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию). Линейный поиск заданного значения в массиве. Алгоритмы работы с элементами массива с однократным просмотром массива. Сортировка одномерного массива. Простые методы сортировки (метод пузырька, метод выбора, сортировка вставками). Сортировка слиянием. Быстрая сортировка массива (алгоритм QuickSort). Двоичный поиск в отсортированном массиве.

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива; умение использовать в программах данные различных типов

с учётом ограничений на диапазон их возможных значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; умение использовать средства отладки программ в среде программирования.

Проверяемые метапредметные результаты: базовые исследовательские действия, самоорганизация, самоконтроль.

Что нужно знать:

как организовать простой перебор данных при проверке условий без оптимизации вычислений;

как обрабатывать данные с помощью электронных таблиц или собственной программы, как правило, написать правильную программу значительно проще;

как реализовать на языке программирования общую структуру цикла перебора для подсчета значений, удовлетворяющих заданным условиям.

Пример общей структуры цикла (Python):

```
count = 0
for n in range(a, b+1):
    if условие выполнено:
        count += 1
print( count )
```

Pascal:

```
count := 0;
for n:=a to b do
    if условие выполнено then
        count := count + 1;
writeln(count);
```

C++:

```
int count = 0;
for(int n = a; n <= b; n++)
```

```

if( условие выполнено )
    count += 1;
std::cout << count;

```

как проверить делимость числа n на число d с помощью операции взятия остатка от деления n на d : если остаток равен 0, число n делится на d нацело

проверка на языке Python выглядит так:

```

if n % d == 0:
    print("Делится")
else: print("Не делится")

```

тоже самое на Pascal

```

if n mod d = 0 then
    writeln('Делится')
else writeln('Не делится')

```

то же самое на C++

```

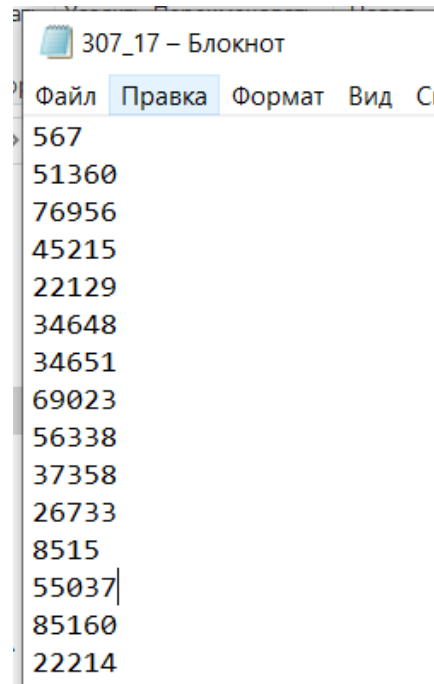
if( n % d == 0 )
    std::cout << "Делится";
else std::cout << "Не делится";

```

Пример формулировки задания (используется открытый вариант №307)

В файле содержится последовательность натуральных чисел. Её элементы могут принимать целые значения от 1 до 100 000 включительно. Определите количество пар последовательности, в которых сумма остатков от деления обоих элементов на 12 равна минимальному элементу последовательности. В ответе запишите количество найденных пар, затем **минимальную** из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Пример исходных данных в файле:



```

307_17 - Блокнот
Файл  Правка  Формат  Вид  Службы
567
51360
76956
45215
22129
34648
34651
69023
56338
37358
26733
8515
55037
85160
22214

```

Решение (Python)

Алгоритм:

1. Чтение данных – файл целиком считывается, все числа преобразуются в список целых чисел.
2. Поиск минимума – находится минимальное значение среди всех элементов последовательности (min_value).
3. Инициализация – счётчик подходящих пар ($\text{count} = 0$) и переменная для минимальной суммы элементов пары ($\text{min_sum} = \infty$).
4. Перебор пар – для каждого индекса i от 0 до $\text{len}(\text{numbers})-2$:
 берутся два соседних элемента $\text{numbers}[i]$ и $\text{numbers}[i+1]$
 вычисляется сумма их остатков от деления на 12
 если эта сумма равна min_value , то:
 счётчик увеличивается на 1
 сумма элементов пары сравнивается с текущим min_sum , и если она меньше, то min_sum обновляется
5. Результат – возвращаются два числа: количество подходящих пар и минимальная сумма элементов среди

них.

Реализация (Python):

```
def find_pairs(filename):
    with open(filename, 'r') as file:
        numbers = list(map(int, file.read().split()))

    min_value = min(numbers)
    count = 0
    min_sum = float('inf')

    for i in range(len(numbers) - 1):
        a, b = numbers[i], numbers[i+1]
        if (a % 12) + (b % 12) == min_value:
            count += 1
            min_sum = min(min_sum, a + b)

    return count, min_sum
```

```
filename = input("Введите название файла: ")
result = find_pairs(filename)
print(*result)
```

Можно выполнить задание средствами электронных таблиц. Алгоритм работы тот же, расчеты выполняются с помощью формул, по примеру выполнения задания № 9.

Анализ ошибок

Типичные ошибки обучающихся при выполнении данного задания аналогичны ошибкам, описанным в задании № 9, кроме этого добавились ошибки, связанные с владением языком программирования (если решение выполнено с

помощью программирования).

1. Неправильный перебор пар – берут всевозможные комбинации элементов (for i in range(n): for j in range(n)...) вместо только соседних, что приводит к колоссальному завышению количества и полному несоответствию условию о «двух идущих подряд элементах».

2. Путаница с остатками – находят остаток от деления минимального элемента на 12 вместо того, чтобы брать остатки от элементов пары; или путают $a \% 12$ с $12 \% a$, что меняет смысл.

3. Ошибка в поиске минимума – ищут минимум только среди элементов, удовлетворяющих какому-то условию, хотя в задаче требуется глобальный минимум всей последовательности.

4. Неверная инициализация min_sum – ставят начальное значение 0, из-за чего минимальная сумма никогда не обновится (так как все суммы положительны и больше нуля); правильно использовать float('inf') или None.

5. Смешение типов данных – пытаются записать ответ как одно число, хотя по условию нужно вывести два числа через пробел.

6. Проблемы с чтением файла – не учитывают пустые строки в конце, забывают про strip(), пытаются читать файл несколько раз без закрытия или без seek(0).

7. Игнорирование границ – при переборе используют range(len(numbers)) вместо range(len(numbers)-1), что вызывает ошибку IndexError при обращении к numbers[i+1].

8. Неоптимальность – хранят все числа, хотя можно было бы обрабатывать их на лету, но для данной задачи это не критично.

9. Отсутствие проверки на пустой файл – если файл пуст или содержит менее двух чисел, программа падает с ошибкой, хотя должна обрабатывать крайние случаи.

Несформированные метапредметные умения	Проявление в типичных ошибках
Базовые исследовательские действия (1.2.4, 1.2.5, 1.2.7)	Неумение организовать простой перебор данных. Отсутствие навыка обработки данных с помощью электронных таблиц или программы. Неумение реализовать общую структуру цикла перебора.
Регулятивные УУД (Самоорганизация) (3.1)	Неумение составить план решения: чтение данных → поиск минимума → перебор пар → подсчёт.
Регулятивные УУД (Самоконтроль) (3.2)	Неумение проверить границы циклов. Отсутствие проверки крайних случаев (пустой файл, менее двух чисел).
Работа с информацией (1.3.2., 1.3.4.)	Неумение протестировать на малом примере.

Несформированные метапредметные умения	Проявление в типичных ошибках
	Неумение правильно прочитать данные из файла. Ошибки при работе со строками и преобразованием типов.

Способы избежать таких ошибок:

Внимательное чтение условия – выделяйте ключевые слова («поряд», «идущих подряд», «минимальный элемент всей последовательности») и переформулируйте задачу своими словами, чтобы исключить неверную трактовку.

Тестирование на малом примере – перед запуском на больших данных вручную просчитайте 5–10 чисел, сравните с ожидаемым ответом; это выявляет ошибки в логике и границах циклов.

Пошаговая отладка с `print()` – выводите промежуточные значения: минимум, остатки для каждой пары, сумму, счётчик – на небольшом фрагменте данных.

Использование осмысленных имён – называйте переменные `pair`, `sum_remainders`, `min_pair_sum`, а не `x`, `y`, `z` – это упрощает чтение и предотвращает путаницу.

Помнить о правильной инициализации `min_sum = float('inf')` – гарантирует корректное обновление, в отличие от 0 или `None`.

Проверка крайних случаев – добавьте условия: если чисел меньше 2 → вернуть (0, 0) или сообщение; если файл пуст → обработать исключение.

Перебор только соседних – строго используйте `for i in range(len(numbers)-1)`, никогда не вкладывайте циклы друг в друга для этой задачи.

Однократное чтение файла – читайте один раз и сохраняйте в список; избегайте множественных открытий и `seek()` без необходимости.

Сравнение с эталоном – если есть возможность, решите задачу другим способом (например, вручную для малого набора или с помощью генератора пар) и сравните результаты.

Краткий чек-лист перед отправкой:

- пары только соседние;
- минимум всей последовательности;
- остатки от обоих элементов;
- сумма остатков равна минимуму;
- вывод двух чисел через пробел;
- обработка пустого/короткого файла

Задание 23

Тема: динамическое программирование.

Уровень сложности: повышенный.

Рекомендуемое время выполнения: 8 минут. Умение анализировать ход исполнения алгоритма

Проверяемые элементы содержания: Массивы и последовательности чисел. Вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения, среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию). Линейный поиск заданного значения в массиве. Алгоритмы работы с элементами массива с однократным просмотром массива. Сортировка одномерного массива. Простые методы сортировки (метод пузырька, метод выбора, сортировка вставками). Сортировка слиянием. Быстрая сортировка массива (алгоритм QuickSort). Двоичный поиск в отсортированном массиве.

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Владение универсальным языком программирования высокого уровня (Паскаль, Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции; умение осуществлять анализ предложенной программы: определять результаты работы программы при заданных исходных данных; определять, при каких исходных данных возможно получение указанных результатов; выявлять данные, которые могут привести к ошибке в работе программы; формулировать предложения по улучшению программного кода. Владение методами рекурсивным методом вычислений или динамическим программированием.

Проверяемые метапредметные результаты: базовые исследовательские действия, самоорганизация, самоконтроль.

Что нужно знать:

динамическое программирование – это способ решения сложных задач путем сведения их к более простым задачам того же типа;

с помощью динамического программирования решаются задачи, которые требуют полного перебор вариантов:

«подсчитайте количество вариантов...»

«как оптимально распределить...»

«найдите оптимальный маршрут...»

динамическое программирование позволяет ускорить выполнение программы за счет использования дополнительной памяти; полный перебор не требуется, поскольку запоминаются решения всех задач с меньшими значениями параметров.

Пример формулировки задания (используется открытый вариант №307)

Исполнитель преобразует число на экране. У исполнителя есть две команды, которые обозначены латинскими буквами:

А. Прибавь 1

В. Поменяй местами

Первая из этих команд увеличивает число на экране на 1. Вторая команда применяется только к числу, у которого цифра в разряде десятков по значению меньше цифры, стоящей в разряде единиц, и действует, заменяя число на экране числом, в котором цифры двух младших разрядов поменялись местами. Программа для исполнителя – это последовательность команд. Сколько существует программ, для которых при исходном числе 100 результатом является число 143? Траектория вычислений программы – это последовательность результатов выполнения всех команд программы.

Например, для программы АВА при исходном числе 13 траектория состоит из чисел 14, 41, 42.

Решение (Python)

```
from functools import lru_cache
def B(x): #выполняется команда В
    y=str(x)
    y=y[:-2]+y[-1]+y[-2]
    return int(y)
```

```
@lru_cache
def f(a,k):
    if a==k: return 1
    if a>k : return 0
    s=str(a)
    if len(s) >= 2 and s[-2] < s[-1]:
```

```

    return f(a+1,k)+f(B(a),k)
else:
    return f(a+1,k)
print(f(100,143))

```

Опишем алгоритм:

1. Функция $B(x)$ – меняет местами две последние цифры числа x :

- преобразует число в строку
- берёт всё, кроме двух последних цифр, затем добавляет последнюю цифру, затем предпоследнюю
- например: $1234 \rightarrow 1243$

2. Рекурсивная функция $f(a, k)$ – подсчитывает количество способов добраться от числа a до числа k , используя две операции:

- $a + 1$ – увеличить на 1 (всегда доступна)
- $B(a)$ – поменять местами две последние цифры (доступна только если предпоследняя цифра меньше последней, т.е. $s[-2] < s[-1]$)

3. Базовые случаи:

- если $a == k \rightarrow$ возвращает 1 (способ найден)
- если $a > k \rightarrow$ возвращает 0 (перебор, дальше идти нельзя)

4. Рекурсивный переход:

- если последние две цифры образуют возрастающую пару (предпоследняя $<$ последней), то можно пойти двумя путями: $a+1$ и $B(a)$

- иначе – только $a+1$

5. Кеширование (`@lru_cache`) – запоминает результаты для уже вычисленных пар (a, k) , чтобы не пересчитывать их повторно, что критически важно для рекурсии с большим числом состояний.

6. Вызов $f(100, 143)$ – вычисляет количество способов преобразовать число 100 в число 143 с помощью описанных правил.

Можно выполнить задание динамическим программированием, обычно этот метод применяется, если очень глубокая рекурсия.

```

def B(x):
    «»"Меняет местами две последние цифры числа»»"
    s = str(x)
    return int(s[:-2] + s[-1] + s[-2])

def solve_dp(start, target):
    «»"
    Динамическое программирование для подсчёта количества способов
    преобразовать число start в число target.

    dp[i] = количество способов достичь числа i из start
    «»"
    # Инициализация массива dp размера target + 1
    dp = [0] * (target + 1)
    dp[start] = 1 # из начального числа есть 1 способ (стоять на месте)

    # Перебираем все числа от start до target
    for a in range(start, target + 1):
        if dp[a] == 0:
            continue # если число недостижимо, пропускаем

        # Операция 1: a + 1
        if a + 1 <= target:
            dp[a + 1] += dp[a]

        # Операция 2: B(a) (меняем местами последние цифры)
        # Доступна только если предпоследняя цифра меньше последней
        s = str(a)
        if len(s) >= 2 and s[-2] < s[-1]:

```

```

    b = B(a)
    if b <= target:
        dp[b] += dp[a]

return dp[target]

# Основная программа
start = 100
target = 143
result = solve_dp(start, target)int(f"Количество способов: {result}")

```

Анализ ошибок

Типичные ошибки обучающихся при выполнении данного задания.

1. **Неправильное определение порядка обхода** – пытаются заполнять dp слева направо, но не учитывают, что операция $B(a)$ может дать число меньше a , из-за чего возникает зависимость от будущих значений; в данной задаче $B(a)$ почти всегда больше a (кроме случаев с нулём), но об этом нужно помнить.
2. **Ошибка в индексации** – создают массив размером $target+1$, но не проверяют, что $B(a) \leq target$, что приводит к `IndexError` при обращении к `dp[b]`.
3. **Путаница с направлением перехода** – в ДП "снизу вверх" нужно идти от `start` до `target`, а не наоборот, иначе теряется смысл накопления количества способов.
4. **Забывают про условие доступности операции** – применяют $B(a)$ для всех чисел, хотя по условию она доступна только если предпоследняя цифра меньше последней ($s[-2] < s[-1]$), иначе ответ становится неверным.
5. **Игнорирование начального состояния** – забывают инициализировать `dp[start] = 1`, из-за чего все значения остаются нулями и ответ всегда 0.
6. **Неверная логика накопления** – используют присваивание `dp[a+1] = dp[a]` вместо сложения `dp[a+1] += dp[a]`, что теряет информацию о других путях к этому числу.
7. **Проблемы с преобразованием строк** – при работе с $B(x)$ не учитывают, что число может стать короче (например, $B(10) = "01" \rightarrow 1$), и не проверяют длину строки перед обращением к `s[-2]`.
8. **Ошибка в условии сравнения** – путают $s[-2] < s[-1]$ с $s[-2] > s[-1]$ или используют `<=` вместо `<`, что меняет

множество доступных переходов.

9. **Неправильное использование кеша** – в рекурсивном варианте забывают декоратор `@lru_cache`, из-за чего программа работает экспоненциально долго и переполняет стек.

10. **Отсутствие проверки на выход за границы** – в рекурсии забывают базовый случай `if a > k: return 0`, из-за чего рекурсия уходит в бесконечность.

11. **Смешение *start* и *target*** – в рекурсивной функции `f(a, k)` путают, где начальное число, а где конечное, особенно при вызове `f(100, 143)` – передают аргументы в неправильном порядке.

12. **Непонимание сути ДП** – пытаются решить задачу полным перебором всех вариантов без мемоизации, что приводит к катастрофическому росту времени (экспонента) и переполнению стека.

13. **Ошибки в типах** – при работе с `B(x)` не преобразуют строку обратно в число через `int()`, из-за чего сравнивают строки с числами или выполняют арифметику над строками.

14. **Не тестируют на малых значениях** – запускают программу сразу на больших числах ($100 \rightarrow 143$), не проверив на примерах типа $10 \rightarrow 15$, где легко просчитать руками и увидеть ошибку.

15. **Путаница между количеством способов и последовательностью шагов** – пытаются вывести сами пути, а не их количество, хотя задача требует только числа способов.

Несформированные метапредметные умения	Проявление в типичных ошибках
Базовые исследовательские действия (1.2.4, 1.2.5, 1.2.7)	Неумение организовать рекурсивный или итеративный перебор. Отсутствие навыка использования динамического программирования (мемоизация). Неумение определить порядок обхода состояний.
Регулятивные УУД (Самоорганизация) (3.1)	Неумение составить рекурсивную функцию с базовыми случаями. Отсутствие навыка работы с кешированием (<code>@lru_cache</code>).
Регулятивные УУД (Самоконтроль) (3.2)	Неумение проверить границы рекурсии. Отсутствие тестирования на малых значениях. Неумение выполнить ручной подсчёт для проверки.
Базовые логические действия (1.1.3., 1.1.4.)	Неумение определить условие доступности операции <code>B</code> . Путаница с направлением перехода в ДП.

Способы избежать таких ошибок:

Чёткое разделение этапов – всегда фиксируйте на бумаге: начальное состояние (`dp[start]=1`), допустимые переходы (условие для `B`), направление обхода (от `start` до `target`), чтобы не путать логику.

Обязательная проверка границ – перед любым обращением к `dp[b]` добавляйте условие `if b <= target`, а при работе со строками – `if len(s) >= 2`, чтобы исключить вылеты.

Использование отладочного вывода – выводите массив `dp` на каждом шаге для малого примера (`start=10`, `target=15`) и сравнивайте с ручным подсчётом – это мгновенно выявляет ошибки в переходах.

Применение `+=` вместо `=` – всегда накапливайте количество способов через сложение, так как в число можно прийти разными путями; присваивание затрёт предыдущие результаты.

Тестирование на граничных случаях – проверяйте `start == target` (ответ 1), числа с нулём на конце (`B(10)=1`), и случаи, где операция `B` недоступна (например, 11, 32) – это покрывает все возможные сценарии.

Ручной просчёт для малых данных – возьмите 3–5 чисел и пошагово просчитайте все переходы, затем сравните с результатом программы – это надёжнее любых абстрактных рассуждений.

Использование `assert` для самопроверки – добавьте в код проверки: `assert dp[start] == 1`, `assert all(v >= 0 for v in dp)` – они сработают при нарушении инвариантов.

Чёткое именование – называйте переменные `start`, `target`, `dp`, `ways` вместо `a`, `k`, `f` – это снижает путаницу и делает код самодокументируемым.

Постепенное усложнение – сначала реализуйте только операцию `+1`, убедитесь, что работает, затем добавьте `B`, проверяя изменения пошагово.

Фиксация условия в комментарии – прямо в коде пишите комментарий: `# B доступна только если предпоследняя цифра < последней`, чтобы не забыть при отладке.

Единый подход – выберите один метод (рекурсия с кешем или итеративное ДП) и строго его придерживайтесь, не смешивая оба, чтобы избежать логических конфликтов.

Проверка типов – убедитесь, что `B(x)` всегда возвращает `int`, а не строку, особенно при работе с числами типа `10 → "01" → 1`.

Использование готовых шаблонов – для задач на ДП держите под рукой заготовку: инициализация, цикл по состояниям, переходы, вывод – это уменьшает количество случайных ошибок.

Использовать чек-лист перед отправкой, например:

- `dp[start] = 1` инициализирован
- Цикл идёт от `start` до `target` (не наоборот)
- Проверка `b <= target` перед обращением к `dp[b]`
- Условие `s[-2] < s[-1]` (строго меньше)

- Накопление через +=, а не присваивание
- Для рекурсии: базовые случаи $a == k$ и $a > k$
- Декоратор @lru_cache для рекурсивной версии
- Тест на малом примере (10, 15)

○ *Предположение о достигнутых и недостигнутых метапредметных результатах ФГОС*

На основе анализа выполнения заданий участниками ЕГЭ с низким уровнем подготовки (группа «от минимального до 60 т.б.») можно сделать следующие предположения о достигнутых и недостигнутых метапредметных результатах ФГОС.

1. Достигнутые метапредметные результаты

1.1. Работа с информацией

Можно предположить, что достигнуты следующие метапредметные результаты ФГОС из группы «Работа с информацией»:

Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления (таблицы и графы).

Данный результат проявляется в достаточно успешном выполнении заданий № 1, № 10, а также № 2, № 3, № 4, № 7 (базового уровня). Обучающиеся демонстрируют умение извлекать информацию из наглядных источников (графы, схемы, таблицы), работать с готовыми базами данных и текстовыми документами. Однако при усложнении условий (задания № 8, № 9) этот навык оказывается недостаточно сформированным.

Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач (при работе с текстовыми редакторами и электронными таблицами).

Этот результат подтверждается выполнением заданий № 10 (поиск в текстовом процессоре) и частично – заданий № 3, № 18 (работа с базами данных и электронными таблицами). Обучающиеся владеют базовыми навыками работы с программным обеспечением, но испытывают затруднения при необходимости создания сложных каскадных формул (задание 9).

1.2. Базовые логические действия (частично)

Можно предположить частичную сформированность следующих умений:

Устанавливать существенный признак или основания для сравнения, классификации и обобщения.

Данное умение проявляется в успешном решении заданий № 2, № 3, № 4, № 7, где требуется выделить существенные признаки объектов (логические переменные, ключевые поля баз данных, кодовые комбинации, формулы для расчёта объёма информации). Однако при усложнении условий (задания № 8, № 14, № 15) обучающиеся теряют способность выделять существенные признаки.

Выявлять закономерности и противоречия в рассматриваемых явлениях.

Данное умение частично сформировано, что подтверждается выполнением заданий № 5, № 6, № 12 (в части, где требуется выполнить алгоритм по шагам или выполнить трассировку). Однако при необходимости самостоятельного выявления закономерностей (задания № 8, № 14, № 16) большинство обучающихся испытывают затруднения.

2. Недостигнутые метапредметные результаты

2.1. Базовые логические действия

Не достигнуты следующие метапредметные результаты из группы «Базовые логические действия»:

2.1.1. Устанавливать существенный признак или основания для сравнения, классификации и обобщения

В заданиях № 8, № 14 обучающиеся не могут определить правильный алфавит системы счисления, путают порядок символов, не видят основания для группировки разрядов. В задании № 15 не могут выделить критические точки, где выражение становится ложным. В задании № 23 не могут определить условие доступности операции.

2.1.2. Выявлять закономерности и противоречия в рассматриваемых явлениях

В заданиях № 5, № 6 обучающиеся не видят закономерности в преобразовании чисел или движении исполнителя. В задании № 8 не видят закономерность в нумерации слов как чисел в системе счисления. В задании № 16 не могут выявить закономерность рекуррентных соотношений. В задании № 23 не могут определить закономерность перехода между состояниями.

2.1.3. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне

В заданиях № 5, № 6, № 8, № 9 обучающиеся не могут самостоятельно сформулировать план решения, выделить все этапы. В задании № 14 не могут определить, какой способ решения выбрать (ручной счёт, программирование, ЭТ). В задании № 17 не могут сформулировать алгоритм обработки данных.

2.2. Базовые исследовательские действия

Не достигнуты следующие метапредметные результаты из группы «Базовые исследовательские действия»:

2.2.1. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения

В заданиях № 5, № 6 обучающиеся не могут выдвинуть гипотезу о том, как будет меняться число или положение исполнителя. В задании № 8 не могут предположить, каким будет номер искомого слова. В задании № 14 не могут

выдвинуть гипотезу о значении суммы. В задании № 15 не могут предположить, какое значение A может быть максимальным.

2.2.2. Находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения

В заданиях № 5, № 6, № 8 обучающиеся не могут обосновать выбор способа решения (ручной счёт vs программирование vs ЭТ). В задании № 9 не могут аргументировать выбор формул для проверки условий. В задании № 17 не могут обосновать границы перебора.

2.2.3. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность

В заданиях № 8, № 14, № 15, № 23 обучающиеся не проверяют полученные результаты, не выполняют обратную проверку, не оценивают достоверность ответа. В задании № 17 не проверяют, что пары являются соседними. В задании № 9 не проверяют правильность составленных формул.

2.3. *Регулятивные УУД (Самоорганизация и Самоконтроль)*

Не достигнуты следующие метапредметные результаты из группы «Регулятивные УУД»:

2.3.1. Самоорганизация: самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи. В заданиях № 5, № 6, № 8, № 9, № 14, № 17, № 23 обучающиеся не могут самостоятельно составить план решения, определить последовательность шагов. Отсутствие системного подхода приводит к пропуску этапов (забывают $N \neq \text{base}$ в цикле, не проверяют границы, не учитывают все условия).

2.3.2. Самоконтроль: давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям. В заданиях № 5, № 8, № 14, № 15, № 17, № 23 обучающиеся не проверяют промежуточные результаты, не выполняют обратную проверку, не тестируют решение на малых примерах. Отсутствие самоконтроля ведёт к накоплению ошибок на каждом этапе решения.

2.4. *Работа с информацией*

Не достигнуты следующие метапредметные результаты из группы «Работа с информацией»:

2.4.1. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач (при работе с табличными редакторами). В задании № 9 обучающиеся не могут составить каскад формул в электронных таблицах для проверки нескольких условий одновременно. Неумение формализовать задачу и разбить её на подзадачи приводит к ошибкам в моделировании.

2.4.2. Владеть навыками получения информации из источников разных типов. В задании № 17 обучающиеся не могут правильно прочитать данные из файла, допускают ошибки при преобразовании типов данных, не учитывают пустые строки и формат входных данных.

ВЫВОД

На основании анализа выполнения заданий группой участников «от минимального до 60 т.б.» можно констатировать, что:

Достигнуты на достаточном уровне:

навыки получения информации из наглядных источников (графы, схемы, таблицы) – задания № 1, № 10;

базовые навыки работы с текстовыми редакторами и простыми функциями электронных таблиц – задания № 10, № 3, № 18;

Частично – умение устанавливать существенные признаки в простых стандартных задачах – задания № 2, № 3, № 4, № 7.

Не достигнуты или сформированы на недостаточном уровне:

базовые логические действия – неумение выявлять закономерности, устанавливать причинно-следственные связи, самостоятельно формулировать проблему в заданиях повышенной сложности (№ 8, 14, 15, 23);

базовые исследовательские действия – неумение выдвигать гипотезы, анализировать достоверность результатов, выбирать эффективный способ решения (№ 5, 6, 14, 15, 17, 23);

регулятивные УУД (самоорганизация и самоконтроль) – отсутствие навыков планирования решения, проверки промежуточных результатов, тестирования на малых примерах (все задания повышенного уровня);

работа с информацией – неумение формализовать задачу, создавать каскадные формулы в электронных таблицах, работать с файловыми данными (№ 9, 17).

Основная причина низких результатов в группе «от минимального до 60 т.б.» связана не столько с отсутствием предметных знаний, сколько с несформированностью метапредметных умений – способности планировать решение, моделировать задачу, контролировать свои действия и проверять результаты. Именно эти дефициты должны стать основным направлением коррекционной работы с данной группой обучающихся.

3.2.2.3. Рекомендации для учителей по совершенствованию преподавания учебного предмета группе обучающихся с низким уровнем подготовки

На основе анализа выполнения заданий ЕГЭ по информатике обучающимися с низким уровнем подготовки (группа «от минимального до 60 т.б.») можно сформулировать следующие рекомендации к их подготовке.

Рекомендация 1: Сфокусироваться на освоении «зон роста» – заданий, имеющих наибольший потенциал для набора баллов

Обучающиеся этой группы уже успешно решают задания № 1, 2, 3, 4, 7, 10, 12, 18, 21 (базового и повышенного уровня), но имеют не очень высокие результаты по заданиям № 5, 6, 8, 9, 11, 13, 16, 19, которые имеют чёткие алгоритмы решения и высокий потенциал для освоения (разрыв с группой «61–80» составляет от 27 до 52 п.п.). Эти задания могут стать основой для набора дополнительных баллов.

Рекомендация в работе с обучаемыми (методика «Приоритетные зоны роста»):

Разработайте для этой категории обучающихся индивидуальные или групповые образовательные маршруты, сфокусированные на отработке конкретного набора заданий, которые в сумме дадут значительный прирост баллов. Освойте следующие темы в порядке приоритета:

Задание 19 (Базовый уровень) – «Анализ алгоритма логической игры (выигрышная позиция за один ход)».

Почему в приоритете: Это базовое задание, которое по логике проще, чем №20 и №21. Группа «61–80» решает его на 83%. Отработка: определение выигрышных и проигрышных позиций за один ход, построение таблицы для всех возможных значений S .

Задание 16 (Повышенный уровень) – «Вычисление рекуррентных выражений».

Почему в приоритете: Колоссальный разрыв между группами (49 п.п.). Задание решается простым последовательным вычислением. Отработка: алгоритм вычисления $F(1), F(2), \dots, F(N)$, использование электронных таблиц для автоматизации.

Задание 6 (Базовый уровень) – «Определение результатов работы алгоритмов управления исполнителями (Черепаха)».

Почему в приоритете: Задание кажется знакомым (лишь 1/5 не приступают), но правильный ответ дают только 54%. Отработка: пошаговое моделирование движения, построение эскиза фигуры, расчёт целочисленных точек.

Задание 8 (Базовый уровень) – «Кодирование данных, комбинаторика, системы счисления».

Почему в приоритете: Разрыв 51 п.п. Отработка: восприятие списка слов как чисел в системе счисления, формула номера слова, учёт чётности и наличия букв.

Задание 13 (Повышенный уровень) – «Умение использовать маску подсети».

Почему в приоритете: Разрыв 36 п.п. Отработка: перевод IP-адресов и масок в двоичную систему, поразрядная конъюнкция, определение адреса сети и номера компьютера.

Задание 11 (Повышенный уровень) – «Подсчёт информационного объёма сообщения».

Почему в приоритете: Разрыв 31 п.п. Отработка: формула Хартли, расчёт $I = K \times \log_2 N$, учёт вероятностного подхода.

Задание 5 (Базовый уровень) – «Формальное исполнение простого алгоритма».

Почему в приоритете: Разрыв 42 п.п. Отработка: перевод чисел между системами счисления, пошаговое выполнение алгоритма, использование таблиц трассировки.

Задание 9 (Базовый уровень) – «Обработка числовой информации в электронных таблицах (сложные условия)».

Почему в приоритете: Самое низкое выполнение (17%), но задание проверяет базовый уровень. Отработка: разбиение условия на логические блоки, пошаговое добавление условий, использование вспомогательных столбцов.

Организация процесса:

Выделите на уроках время для «минутки повторения» по этим темам.

Используйте систему «накопительных карточек», где обучающийся отмечает решенные задачи по каждой теме, переходя к следующей только после успешного выполнения 5-7 однотипных заданий из открытого банка ФИПИ.

Начинайте с базовых заданий (№ 5, 6, 8, 9, 19), затем переходите к повышенному уровню (№ 11, 13, 16). Это позволит постепенно наращивать сложность.

Рекомендация 2: Формировать метапредметные умения через пошаговое решение и моделирование задач

Обучающиеся демонстрируют недостаточную сформированность метапредметных умений (базовых логических и исследовательских действий, регулятивных УУД), что проявляется в неумении выявлять закономерности, строить алгоритмы, анализировать результаты, планировать решение и проверять себя. Особенно ярко это проявляется в заданиях № 5, 6, 8, 9, 14, 15, 17, 23 (страницы 87-106).

Рекомендация (приемы работы):

Задания №5, №6 («Исполнители и алгоритмы»):

Вместо решения сложных задач сразу, практикуйте «ручную прокрутку» (трассировку) алгоритмов на небольших наборах данных.

Научите обучающихся составлять таблицы трассировки, пошагово записывая значения переменных и координаты исполнителя.

Важно: сначала объясняйте на примерах с бумагой и карандашом, и только после этого показывайте решение в среде КуМир или на Python.

Для задания №6: перед вычислениями обязательно делать эскиз фигуры на черновике, отмечая координаты всех поворотов. Это помогает визуализировать пересечение и избежать ошибок с объединением.

Задание 8 («Комбинаторика и системы счисления»):

Для непрограммирующих обучающихся используйте прием «перевод в систему счисления». Научите их воспринимать список слов как числа в системе счисления с основанием, равным количеству букв в алфавите.

Отрабатывайте этот навык на простых примерах (например, слова из 3 букв), постепенно усложняя.

Внедрите обязательный этап «Парсинга задачи»: перед решением выписать «Что дано?», «Что нужно найти?», «Каковы все условия?».

Задание 9 («Электронные таблицы»):

Вместо решения задач целиком, научите разбивать условие на отдельные логические блоки.

Используйте метод «от простого к сложному»: сначала решите задачу, где нужно найти только сумму или только максимум, а затем постепенно добавляйте новые условия.

Например, для нахождения строки с уникальными числами: сначала отработайте формулу СЧЁТЕСЛИ, затем – формулу для проверки равенства, и только потом – объединение этих условий в сложный фильтр.

Отрабатывайте этап формализации: учите выделять этапы построения алгоритма, моделировать, тестировать промежуточные и итоговый результат.

Задание 14 («Позиционные системы счисления»):

Отрабатывайте системный подход: привести все числа к общему основанию → выполнить заёмы (схематично, с таблицей) → перейти к нужной системе счисления (группировка) → найти требуемые цифры → выполнить действие.

Используйте визуализацию: схемы разрядов и цветные таблицы для отслеживания заёмов.

При обучении программированию предупреждайте частые ошибки: не забывать $N \neq 36$ (бесконечный цикл), правильное условие $r > 9$ (не $r \geq 9$), использование `string.digits + string.ascii_uppercase`.

Задание 23 («Динамическое программирование»):

Начните с ручного подсчёта для малых значений (например, $10 \rightarrow 15$), чтобы обучающийся понял логику переходов.

Отрабатывайте базовые случаи рекурсии: `if a == k: return 1, if a > k: return 0`.

Обязательно используйте декоратор `@lru_cache` для кеширования результатов (при решении на Python) или организуйте итеративное заполнение массива DP.

Проверяйте условие доступности операции B: `s[-2] < s[-1]` (строго меньше).

Рекомендация 3: Развивать навыки самоконтроля и регулятивные УУД через систему проверки и рефлексии

Основная масса ошибок в группе возникает из-за отсутствия самоконтроля: обучающийся не проверяют промежуточные результаты, не тестируют решение на малых примерах, не выполняют обратную проверку, не оценивают достоверность ответа по размерности. Особенно ярко это проявляется в заданиях № 5, 8, 14, 15, 17, 23.

Рекомендация (методика «Двойной проверки»):

Внедрите обязательный этап «Проверки решения» после каждого задания. Научите обучающихся использовать следующие приёмы самоконтроля:

Обратная проверка (для заданий № 5, 8, 14):

Если нашли число в системе счисления – переведите его обратно в десятичную систему.

Если нашли номер слова – проверьте, что слово действительно стоит на этом месте.

Для задания №14: проверьте, что сумма цифр не превышает количество цифр $\times 35$ (для 36-ричной системы).

Проверка на граничных условиях (для заданий № 15, 17, 23):

Для задания №15: проверьте найденное A на критической точке ($x=75$).

Для задания №17: проверьте, что пары являются соседними ($\text{range}(\text{len}(\text{numbers})-1)$), а не всевозможными комбинациями.

Для задания №23: протестируйте решение на малых значениях ($\text{start}=10$, $\text{target}=15$) и сравните с ручным подсчётом.

Проверка по размерности (для заданий № 7, 11, 14):

Сумма цифр не может быть больше общего количества цифр $\times (\text{основание}-1)$.

Объём информации не может быть отрицательным.

Проверка с помощью программирования (для всех заданий):

Напишите небольшую программу для проверки ответа (особенно для заданий № 14, 15, 17, 23).

Сравните результат ручного решения с результатом программы.

Чек-лист перед сдачей решения:

Разработайте для обучающихся печатный чек-лист для каждого типа заданий (пример для задания №14):

Я правильно определил основание системы (36)

Я разложил все числа на степени 6

Я правильно перевёл 2023 в 6-ричную систему (13211₆)

Я выполнил все заёмы (проверил каждый шаг)

Я перешёл от 6-ричной к 36-ричной (группировка по 2)

Я посчитал количество разрядов в каждом блоке

Я определил, какие пары дают цифры > 9

Я правильно сложил все цифры > 9

Я проверил ответ альтернативным способом (программой)

Рекомендация 4: Организовать системную работу по формированию навыков алгоритмизации и программирования с использованием разных инструментов

Обучающиеся часто не видят альтернативных способов решения: пытаются решать все задачи вручную, хотя некоторые задания (№ 14, 15, 17, 23) значительно проще и надёжнее решать с помощью программирования или электронных таблиц. При этом они не владеют достаточными навыками программирования для решения задач повышенного уровня сложности.

Рекомендация (системная работа):

Пропедевтика с 7 класса:

Начиная с 7 класса, включайте в уроки задания на формальное исполнение алгоритмов (например, задачи на построение фигур по координатам в графическом редакторе) и на логическое моделирование (задачи на поиск закономерностей). Это заложит базу для метапредметного умения «выявлять закономерности и противоречия».

В 7 классе в разделе «Компьютерная графика» использовать задания на построение чертежей с выделением областей пересечения, объединения фигур (для подготовки к заданию №6).

В 8 классе в разделе «Алгоритмы и программирование» использовать задания не только на разработку алгоритмов, но и на анализ заданного алгоритма (для подготовки к заданиям №5, №6).

Работа с электронными таблицами:

В курсе информатики 9-10 класса усильте практическую часть по работе с электронными таблицами.

Не ограничивайтесь простыми вычислениями, давайте задачи на условные и логические функции (ЕСЛИ, И, ИЛИ, СЧЁТЕСЛИ, СУММЕСЛИ) для фильтрации данных, так как это напрямую связано с алгоритмическим мышлением.

Учите составлять формулы по выделенным в задании условиям, начиная с 9 класса в разделе «Моделирование как метод познания» и возвращаясь к отработке навыков в 11 классе в теме «Анализ данных с помощью электронных таблиц».

Выбор инструмента решения:

Обсуждайте с обучающимися, какой способ решения (программирование или ЭТ) для них проще и быстрее в каждой конкретной задаче.

Предлагайте решать задания разными способами, чтобы обучающийся мог выбрать наиболее комфортный для себя. Это также является элементом формирования метапредметного умения (оценка эффективности способа решения).

Для заданий №14, №15, №17, №23 показывайте как ручное решение, так и программное, чтобы обучающийся видел преимущества каждого подхода.

Программирование на Python:

Дайте обучающимся универсальный шаблон для решения задач на системы счисления (задание 14):

```
python
N = выражение
s = 0
while N > 0:
    r = N % 36
    if r > 9:
        s += r
    N //= 36
print(s)
```

Для задания №15 дайте шаблон с функцией ДЕЛ и перебором.

Для задания №17 дайте шаблон чтения файла и перебора пар.

Для задания №23 дайте шаблон рекурсивной функции с кешированием или итеративного ДП.

Преимущество программ обучения:

Формирование умений, проверяемых на ГИА, должно начинаться с 7 класса и продолжаться систематически на всех этапах обучения.

Включайте задания по типу заданий из КИМ ЕГЭ при изучении соответствующих разделов, усложняя формулировки условий постепенно.

ОБОБЩАЮЩИЙ ВЫВОД:

Акцентирование отработки умений, требуемых в решении указанных 8 тем, в совокупности с 9 успешно выполняемыми заданиями, позволит учащимся из группы «от минимального до 60 т.б.» уверенно набрать 60–70 тестовых баллов. Все задания имеют стандартные, повторяющиеся алгоритмы решения и не требуют глубоких навыков программирования, что делает их реалистичными для освоения при систематической пошаговой отработке.

Основные методические акценты:

трассировка и моделирование (№ 5, 6);

работа с системами счисления и комбинаторика (№ 8, 13);

формализация задач в электронных таблицах (№ 9);

стандартные вычислительные алгоритмы (№ 11, 16);

игровые и логические модели (№ 19);

рекомендуемая последовательность освоения: базовые задания (№ 5, 6, 8, 9, 19) → затем повышенный уровень (№ 11, 13, 16). это позволит постепенно наращивать сложность и формировать уверенность в решении у слабо подготовленных обучающихся.

3.2.3. Методический анализ выполнения заданий обучающимися со средним уровнем подготовки (в группе от 61 до 80 т.б.), включая методические рекомендации для учителей

3.2.3.1. Описание предметной подготовки участников ЕГЭ со средним уровнем подготовки, анализ типичных дефицитов в подготовке

○ *Описание достигнутых предметных результатов ФГОС: освоенных тем программы, сформированных умений и т.п. (если имеются)*

Для данной группы обучаемых не стоит выделять задания базового уровня, так как все участники этой группы выполнили их на высоком уровне (выполнимость 80% и выше), кроме задания, которое отметим в следующем пункте.

Наиболее успешно выполненные задания повышенного уровня сложности: 12, 13, 14, 16, 18, 20.

Наиболее успешно выполненные задания высокого уровня сложности: 21.

С десятью прекрасно выполненными заданиями базового уровня сложности всего получается 17 выполненных заданий, что соответствует 70 тестовым баллам.

Можно констатировать, что обучающимися этой группы на высоком уровне освоены следующие элементы содержания школьной программы по информатике и умения, соответствующие заданиям повышенного уровня:

Задание 12.

Разделы/темы школьной программы: 11 кл., п. 113.7.3. Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат.

Умения и навыки: Понимание базовых алгоритмов обработки числовой и текстовой информации (запись чисел в позиционной системе счисления, делимость целых чисел; нахождение всех простых чисел в заданном диапазоне; обработка многоразрядных целых чисел; анализ символьных строк и других), алгоритмов поиска и сортировки.

Метапредметные результаты: *Базовые логические действия* (Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения.). *Базовые исследовательские действия* (Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях.).

Задание 13

Элементы содержания: Сетевые протоколы. Сеть Интернет. Адресация в сети Интернет. Разделение IP-сети на подсети с помощью масок подсетей.

Умения и навыки: Наличие представлений о базовых принципах организации и функционирования компьютерных сетей. Умение использовать маску подсети.

Метапредметные результаты: *Работа с информацией* (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.) *Коммуникативные УУД* (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств). *Коммуникативные УУД* (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств).

Задание 14

Элементы содержания: Системы счисления. Развёрнутая запись целых и дробных чисел в позиционной системе счисления. Свойства позиционной записи числа: количество цифр в записи, признак делимости числа на основание системы счисления. Алгоритм перевода целого числа из Р-ичной системы счисления в десятичную. Алгоритм перевода целого числа из десятичной системы счисления в Р-ичную. Перевод конечной десятичной дроби в Р-ичную. Двоичная, восьмеричная и шестнадцатеричная системы счисления, связь между ними. Арифметические операции в позиционных системах счисления.

Умения и навыки: Умение использовать при решении задач свойства позиционной записи чисел, алгоритмы по-

строения записи числа в позиционной системе счисления с заданным основанием и построения числа по строке, содержащей запись этого числа в позиционной системе счисления с заданным основанием; умение выполнять арифметические операции в позиционных системах счисления.

Метапредметные результаты: *Базовые логические действия* (Устанавливать существенный признак или основания для сравнения, классификации и обобщения. Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности. Развивать креативное мышление при решении жизненных проблем).

Задание 16

Элементы содержания: Рекурсия. Рекурсивные процедуры и функции.

Умения и навыки: Вычисление рекуррентных выражений.

Метапредметные результаты: *Базовые логические действия* (Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. *Базовые исследовательские действия* (Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях.).

Задание 18.

Разделы/темы школьной программы: Анализ данных с помощью электронных таблиц. Вычисление суммы, среднего арифметического, наибольшего (наименьшего) значения диапазона. Решение задач оптимизации с помощью электронных таблиц.

Умения и навыки: Умение использовать электронные таблицы для анализа, представления и обработки данных (включая выбор оптимального решения, подбор линии тренда, решение задач прогнозирования); умение использовать табличные (реляционные) базы данных и справочные системы.

Метапредметные результаты: *Работа с информацией* (Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности).

Задание 20

Элементы содержания: Дискретные игры двух игроков с полной информацией. Построение дерева перебора вариантов, описание стратегии игры в табличной форме. Выигрышные и проигрышные позиции. Выигрышные стратегии.

Умения и навыки: Умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде. Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию.

Метапредметные результаты: *Работа с информацией* (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.). *Коммуникативные УУД* (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств).

Задание 21

Разделы/темы школьной программы: Дискретные игры двух игроков с полной информацией. Построение дерева перебора вариантов, описание стратегии игры в табличной форме. Выигрышные и проигрышные позиции. Выигрышные стратегии.

Умения и навыки: Умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде. Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию.

Метапредметные результаты: *Работа с информацией* (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.) *Коммуникативные УУД* (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств).

○ *Анализ типичных дефицитов в предметной подготовке: неосвоенные или слабо освоенные темы программы, несформированные или недостаточно сформированные умения, навыки, виды познавательной деятельности,*

проявляющиеся в типичных ошибках при выполнении соответствующих заданий

Таблица 2-18

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП
1.	Задание 23 <u>Элементы содержания:</u> Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат. <u>Умения и навыки:</u> Владение универсальным языком программирования высокого уровня (Паскаль, Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции; умение осуществлять анализ предложенной программы: определять результаты работы программы при заданных исходных данных; определять, при каких исходных данных возможно получение указанных результатов; выявлять данные, которые могут привести к ошибке в работе программы; формулировать предложения по улучшению программного кода. <u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов). <i>Регулятивные УУД 3.1 Самоорганизация</i> (Самостоятельно осуществлять познава-	Базовый уровень: 11 кл., п. 113.7.3. Углубленный уровень: 10 кл., п. 114.6.3; 11 кл., п. 114.7.2

	тельную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям. Самостоятельно составлять план решения проблемы с учётом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретённый опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний.) 3.2 <i>Самоконтроль</i> (Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям. Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приёмы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению).	
2.	Задание 24 <u>Элементы содержания:</u> Обработка символьных данных. Встроенные функции языка программирования для обработки символьных строк. Алгоритмы обработки символьных строк: подсчёт количества появлений символа в строке, разбиение строки на слова по пробельным символам, поиск подстроки внутри данной строки, замена найденной подстроки на другую строку. Генерация всех слов в некотором алфавите, удовлетворяющих заданным ограничениям. Преобразование числа в символьную строку и обратно. <u>Умения и навыки:</u> Владение универсальным языком программирования высокого уровня (Паскаль, Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции; умение осуществлять анализ предложенной программы: определять результаты работы программы при заданных исходных данных; определять, при каких исходных данных возможно получение указанных результатов; выявлять данные, которые могут привести к ошибке в работе программы; формулировать предложения по улучшению программного кода. <u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых	Базовый уровень: 11 кл. п. 113.7.3. Углубленный уровень: 10 кл., п. 114.6.3; 11 кл., п. 114.7.2

	условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов). <i>Регулятивные УУД 3.1 Самоорганизация</i> (Самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям. Самостоятельно составлять план решения проблемы с учётом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретённый опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний.) <i>3.2 Самоконтроль</i> (Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям. Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приёмы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению).	
3.	Задание 25 <u>Элементы содержания:</u> Алгоритмы обработки натуральных чисел, записанных в позиционных системах счисления: разбиение записи числа на отдельные цифры, нахождение суммы и произведения цифр, нахождение максимальной (минимальной) цифры. Представление числа в виде набора простых сомножителей. Алгоритм быстрого возведения в степень. Поиск простых чисел в заданном диапазоне с помощью алгоритма «решето Эратосфена» <u>Умения и навыки:</u> Умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива; умение использовать в программах	Базовый уровень: 11 кл., п. 113.7.3. Углубленный уровень: 10 кл., п. 114.6.3; 11 кл., п. 114.7.2

	данные различных типов с учётом ограничений на диапазон их возможных значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; умение использовать средства отладки программ в среде программирования. <u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов). <i>Регулятивные УУД 3.1 Самоорганизация</i> (Самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям. Самостоятельно составлять план решения проблемы с учётом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретённый опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний.) <i>3.2 Самоконтроль</i> (Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям. Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приёмы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению).	
--	---	--

4.	Задание 26 <u>Элементы содержания:</u> Вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения, среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию). Линейный поиск заданного значения в массиве. Алгоритмы работы с элементами массива с однократным просмотром массива. Сортировка одномерного массива. <u>Умения и навыки:</u> Умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива; умение использовать в программах данные различных типов с учётом ограничений на диапазон их возможных значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; умение использовать средства отладки программ в среде программирования. <u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной	Базовый уровень: 11 кл., п. 113.7.3. Углубленный уровень: 10 кл., п. 114.6.3; 11 кл., п. 114.7.2
----	--	---

	деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов). <i>Регулятивные УУД 3.1 Самоорганизация</i> (Самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям. Самостоятельно составлять план решения проблемы с учётом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретённый опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний.) <i>3.2 Самоконтроль</i> (Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям. Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приёмы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению).	
5.	Задание 27 <u>Элементы содержания:</u> Анализ данных. Основные задачи анализа данных: прогнозирование, классификация, кластеризация, анализ отклонений. Последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и/или построение модели, преобразование данных, визуализация данных, интерпретация результатов. <u>Умения и навыки:</u> Умение классифицировать основные задачи анализа данных (прогнозирование, классификация, кластеризация, анализ отклонений); понимать последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и/или построение модели, преобразование данных, визуализация данных, интерпретация результатов. <u>Метапредметные результаты:</u> <i>Базовые логические действия</i> (Устанавливать существенный признак или основания для сравнения, классификации и обобщения. Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности. Развивать креативное мышление при решении жизненных проблем).	Базовый уровень: 10 кл., п. 114.6.4 Углубленный уровень: 10 кл., п. 114.6.3; 11 кл., п. 114.7.2

○ *Реалистичные «зоны роста» в части предметной подготовки для уверенного получения 80 и более т.б.*

На основе анализа данных Таблицы 2-13 для группы «от 61 до 80 т.б.» и сравнения с результатами группы «от 81

до 100 т.б.» , выделены 5 заданий, которые представляют собой наиболее реалистичные «зоны роста» для достижения 80–83 тестовых баллов.

Эти задания:

имеют средний или низкий процент выполнения в текущей группе (от 0,96% до 69,62%);

демонстрируют высокий процент выполнения в группе «81–100» (от 60,67% до 99,58%);

имеют чёткий, повторяющийся алгоритм решения;

не включены в список уже успешно решаемых заданий (№ 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 13, 14, 16, 18, 20, 21).

Задание 9 (Базовый уровень)

Тема: Обработка числовой информации в электронных таблицах (сложные условия).

Показатель	Значение
% выполнения в группе «61–80»	69,62%
% выполнения в группе «81–100»	95,82%
Разрыв	26,20 п.п.

Почему это «зона роста»: задание базового уровня, но показывает один из самых низких результатов среди базовых заданий в группе «61–80». При этом задания № 3 и № 18 (также на работу с электронными таблицами) обучающиеся решают очень хорошо. Это говорит о том, что обучающиеся умеют использовать отдельные функции (МАКС, МИН, СРЗНАЧ), но не могут составить каскад формул для проверки нескольких условий одновременно. Именно навык формализации, разбиения задачи на подзадачи и моделирования здесь ключевой. Группа «81–100» решает задание почти на 96%, что подтверждает его доступность при целенаправленной отработке.

Что отрабатывать:

разбиение условия на отдельные логические блоки (например: «все числа различны» → проверка повторов через СЧЁТЕСЛИ);

пошаговое добавление условий: сначала одно, потом два, потом три;

использование вспомогательных столбцов для промежуточных вычислений;

применение функций ЕСЛИ, И, ИЛИ, СЧЁТЕСЛИ, СУММЕСЛИ;

составление каскада формул для проверки выполнения двух и более условий одновременно.

Методический приём: научить обучающихся составлять блок-схему проверки условий на бумаге, а затем переносить её в электронную таблицу.

Задание 11 (Повышенный уровень)

Тема: Подсчёт информационного объёма сообщения.

Показатель	Значение
% выполнения в группе «61–80»	54,81%
% выполнения в группе «81–100»	87,45%
Разрыв	32,64 п.п.

Почему это «зона роста»: задание имеет стандартную формулу расчёта информационного объёма: $I = K \times \log_2 N$. Разрыв между группами составляет более 32 п.п., что говорит о высоком потенциале роста. Ошибки возникают из-за неверного определения мощности алфавита, путаницы с равномерным/неравномерным кодированием или невнимательного прочтения условия. Группа «81–100» решает задание почти на 90%, что подтверждает его доступность при целенаправленной отработке.

Что отрабатывать:

формула Хартли: $I = \log_2 N$ для одного символа;

расчёт общего объёма: $I_{\text{общ}} = K \times \log_2 N$;

учёт вероятностного подхода (если символы неравновероятны);

решение задач на минимальное количество бит для кодирования символа;

перевод единиц измерения информации (биты $\rightarrow$ байты $\rightarrow$ Кбайты).

Задание 15 (Повышенный уровень)

Тема: Знание основных понятий и законов математической логики.

Показатель	Значение
% выполнения в группе «61–80»	50,00%
% выполнения в группе «81–100»	87,03%
Разрыв	37,03 п.п.

Почему это «зона роста»: задание показывает разрыв более 37 п.п. между группами. Обучающиеся из группы «81–100» решают его на 87%, что доказывает принципиальную решаемость. Ошибки связаны с непониманием работы кванторов ($\forall$, $\exists$), неправильным преобразованием импликации ($A \rightarrow B = \neg A \vee B$) и неумением выделить критические точки, где выражение становится ложным.

Что отрабатывать:

определение области истинности предикатов с кванторами;
 преобразование логических выражений с использованием законов де Моргана и замены импликации;
 метод подстановки значений для проверки истинности;
 решение задач на поиск минимального/максимального значения, удовлетворяющего условию;
 нахождение «опасных» точек, где выражение может стать ложным.

Задание 17 (Повышенный уровень)

Тема: Умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования.

Показатель	Значение
% выполнения в группе «61–80»	48,65%
% выполнения в группе «81–100»	94,98%
Разрыв	46,33 п.п.

Почему это «зона роста»: колоссальный разрыв между группами (более 46 п.п.). Задание решается написанием простой программы перебора или использованием электронных таблиц. Группа «81–100» решает задание почти на 95%, что говорит о том, что при наличии навыков программирования задача становится тривиальной. Обучающиеся группы «61–80» часто не умеют организовать правильный перебор (путают соседние пары со всевозможными комбинациями) или допускают ошибки при работе с файлами.

Что отрабатывать:

организация цикла перебора элементов последовательности;
 проверка соседних пар (только идущих подряд);
 работа с остатками от деления (%).

Поиск минимума в последовательности.

Чтение данных из файла (в Python: `open, read().split()`).

Инициализация переменных (например, `min_sum = float('inf')`).

Задание 22 (Повышенный уровень)

Тема: Построение математических моделей для решения практических задач. Многопроцессорные системы.

Показатель	Значение
------------	----------

% выполнения в группе «61–80»	56,15%
% выполнения в группе «81–100»	92,47%
Разрыв	36,32 п.п.

Почему это «зона роста»: задание имеет стандартный алгоритм решения через составление таблиц занятости процессоров. Разрыв между группами составляет более 36 п.п. Ошибки возникают из-за непонимания параллельности выполнения команд, неверного учёта зависимостей между операциями или неправильного определения общего времени выполнения.

Что отрабатывать:

составление таблиц занятости процессоров (по тактам).

различие последовательного и параллельного выполнения.

определение зависимых и независимых команд.

расчёт общего времени выполнения при заданном количестве процессоров.

решение задач на моделирование работы многопроцессорных систем.

ИТОГОВЫЙ СПИСОК элементов содержания и умений (приоритетные темы для подготовки к 80–85 т.б.)

№ задания	Проверяемые элементы содержания	Ключевые умения и навыки
9	Обработка числовой информации в электронных таблицах (сложные условия)	Разбиение задачи на логические блоки. Составление каскада формул с использованием СЧЁТЕСЛИ, ЕСЛИ, И, ИЛИ. Проверка выполнения двух и более условий одновременно. Использование вспомогательных столбцов.
11	Подсчёт информационного объёма сообщения	Применение алфавитного подхода. Расчёт количества информации с учётом вероятности и мощности алфавита. Формула Хартли. Перевод единиц измерения.
15	Основные понятия и законы математической логики	Работа с предикатами и кванторами. Поиск области истинности высказывания с переменными. Преобразование импликации $A \rightarrow B = \neg A \vee B$. Выделение критических точек.
17	Составление алгоритма обработки числовой последовательности (программа 10–15 строк)	Организация цикла перебора. Проверка соседних пар. Работа с остатками от деления. Поиск минимума. Чтение данных из файла. Инициализация переменных.
22	Построение математических моделей для решения практических задач. Многопроцессорные системы	Составление таблиц занятости процессоров. Расчёт времени выполнения при последовательном и параллельном исполнении команд. Определение зависимых и независимых команд.

ВЫВОД:

Освоение указанных 6 тем позволит учащимся из группы «от 61 до 80 т.б.» уверенно набрать 80–85 тестовых баллов. Все задания имеют стандартные, повторяющиеся алгоритмы решения и не требуют углублённых знаний, выходящих за рамки школьной программы. При этом они требуют сформированности навыков программирования и алгоритмического мышления.

Основные методические акценты:

Направление подготовки**Задания**

Формализация и моделирование в ЭТ

№ 9

Вычислительные алгоритмы

№ 11, 17

Логические и математические модели

№ 15, 22

Рекомендуемая последовательность освоения:

Задания с чётким формульным алгоритмом (№ 9, 11, 22).

Задания на логические преобразования (№ 15).

Задания на программирование (№ 17).

Это позволит постепенно наращивать сложность и формировать уверенность в решении у обучающихся со средним уровнем подготовки. Особое внимание следует уделить отработке навыков формализации задач в электронных таблицах (№ 9) и навыков программирования (№ 17), так как эти задания показывают наибольший разрыв между группами и имеют высокий потенциал для набора баллов.

3.2.3.2. Методический анализ качества освоения метапредметных результатов ФГОС участниками ЕГЭ со средним уровнем подготовки

○ *Развернутый комментарий на основе заданий / групп заданий, на успешность выполнения которых могла повлиять достаточная или недостаточная сформированность метапредметных умений, с указанием соответствующих метапредметных умений и их проявления в типичных ошибках в ответах участников экзамена на соответствующие задания*

Для анализа типичных ошибок школьников и их причин рассмотрим наиболее unsuccessfully выполненные задания в группе обучающихся с минимальным уровнем подготовки, разберем примеры правильных решений и сопоставим ошибки с соответствующими метапредметными компетенциями.

Наиболее unsuccessfully выполненное задание базового уровня №9 разобрано подробно в п. 3.2.1.2.

Наиболее unsuccessfully выполненное задание повышенного уровня 23 разобрано подробно в п. 3.2.2.2.

Разберем наиболее unsuccessfully задания высокого уровня 24, 25, 26, 27.

Задание 24

Тема: Обработка символьных строк.

Уровень сложности: высокий.

Рекомендуемое время выполнения: 18 минут.

Умение создавать собственные программы (10–20 строк) для обработки символьной информации.

Проверяемые элементы содержания: Обработка символьных данных. Встроенные функции языка программирования для обработки символьных строк. Алгоритмы обработки символьных строк: подсчёт количества появлений символа в строке, разбиение строки на слова по пробельным символам, поиск подстроки внутри данной строки, замена найденной подстроки на другую строку. Генерация всех слов в некотором алфавите, удовлетворяющих заданным ограничениям. Преобразование числа в символьную строку и обратно.

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Владение универсальным языком программирования высокого уровня (Паскаль, Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции; умение осуществлять анализ предложенной программы: определять результаты работы программы при заданных исходных данных; определять, при каких исходных данных возможно получение указанных результатов; выявлять данные, которые могут привести к ошибке в работе программы; формулировать предложения по улучшению программного кода.

Проверяемые метапредметные результаты: базовые исследовательские действия, самоорганизация, самоконтроль.

Что нужно знать:

сначала нужно прочитать строку из файла; эта задача в разных языках программирования решается по-разному; в языке Python удобнее всего использовать такую конструкцию:

```
with open("k7.txt", "r") as F:
    s = F.readline()
```

конструкция with-as – это контекстный менеджер, в данном случае он открывает указанный файл в режиме чтения (второй аргумент «r» при вызове функции open), записывает ссылку на него в файловую переменную F, выполняет тело блока (читает первую строку файла в переменную s) и закрывает (освобождает) файл

в языке PascalABC.NET можно выполнить перенаправление потока ввода:

```
assign( input, 'k7.txt' );
readln(s);
```

программа будет «думать», что читает данные, введенные с клавиатуры (с консоли), а на самом деле эти данные будут прочитаны из файла k7.txt

в языке FreePascal также можно выполнить перенаправление потока ввода, но нужно дополнительно открывать входной поток:

```
assign( input, 'k7.txt' );
reset( input );
readln(s);
```

при работе в среде FreePascal нужно убедиться, что в параметрах компилятора включена поддержка длинных символьных строк; на всякий случай стоит добавить в первой строке программы директиву

```
{ $H+ }
```

Среда PascalABC НЕ ПОДДЕРЖИВАЕТ работу с длинными символьными строками, поэтому для решения задачи использовать версию PascalABC.NET.

В языке C++ используем потоки:

```
#include <fstream>
#include <string>
using namespace std;
int main()
{
    ifstream F(«k7.txt»);
    string s;
    getline( F, s );
```

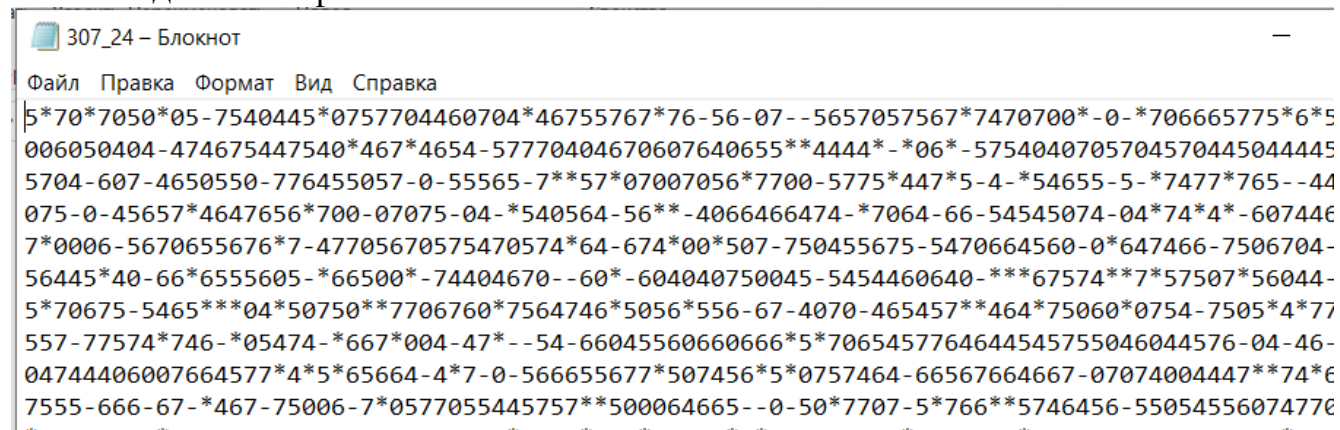
...
}

Для выполнения заданий требуется уметь определять длину строки, обращаться к символу по индексу, копировать и вставлять строки и подстроки.

Пример формулировки задания (используется открытый вариант №307)

Текстовый файл состоит из цифр 0, 4, 5, 6, 7 и знаков арифметических операций «—» и «*» (вычитание и умножение). Определите максимальное количество символов в непрерывной последовательности, которая является корректным арифметическим выражением с целыми неотрицательными числами. В этом выражении никакие два знака арифметических операций не стоят рядом, в записи чисел отсутствуют незначащие (ведущие) нули и число 0 не имеет знака. В ответе укажите количество символов.

Пример представления данных в файле:



```

307_24 - Блокнот
Файл Правка Формат Вид Справка
5*70*7050*05-7540445*0757704460704*46755767*76-56-07--5657057567*7470700*-0-*706665775*6*5
006050404-474675447540*467*4654-57770404670607640655**4444*-*06*-5754040705704570445044445
5704-607-4650550-776455057-0-55565-7**57*07007056*7700-5775*447*5-4-*54655-5-*7477*765--44
075-0-45657*4647656*700-07075-04-*540564-56**-4066466474-*7064-66-54545074-04*74*4*-60744€
7*0006-5670655676*7-47705670575470574*64-674*00*507-750455675-5470664560-0*647466-7506704-
56445*40-66*6555605-*66500*-74404670--60*-604040750045-5454460640-***67574**7*57507*56044-
5*70675-5465***04*50750**7706760*7564746*5056*556-67-4070-465457**464*75060*0754-7505*4*77
557-77574*746-*05474-*667*004-47*- -54-66045560660666*5*7065457764644545755046044576-04-46-
04744406007664577*4*5*65664-4*7-0-566655677*507456*5*0757464-66567664667-07074004447**74*€
7555-666-67-*467-75006-7*0577055445757**500064665--0-50*7707-5*766**5746456-5505455607477€
  
```

Решение (Python)

Для решения этой задачи удобно использовать регулярные выражения в Python.

Шаги решения:

1. Определим шаблон для целого неотрицательного числа. Число может быть отдельным нулём (0) или последовательностью цифр, которая не начинается с нуля ([4567][04567]*). В регулярном выражении это выглядит так: `0|([4567][04567]*)`.

2. Построим шаблон для всего выражения. Корректное выражение – это либо одно число, либо число, за которым следует операция (- или *) и ещё одно корректное выражение. Это записывается как: `(?={число}([-*]{число}+))`.
3. Найдём все совпадения в тексте. Используем `finditer` из модуля `re` для поиска всех подходящих подстрок.
4. Выберем самую длинную подстроку. Среди всех найденных подстрок берём ту, у которой больше всего символов, и возвращаем её длину.

```
from re import finditer
text = open('307_24.txt').readline()
nat = r'(0|([4567][04567]*))' # Шаблон для целого неотрицательного числа
reg = rf'(?=({nat}([-*]{nat}+)))' # Регулярное выражение для корректного выражения
max_len = max(len(match.group(1)) for match in finditer(reg, text)) # Находим все совпадения и выбираем максимальную длину
print(max_len)
```

Важный нюанс: в начале строки добавляется просмотр вперёд (?) – это нужно, чтобы избежать ложных совпадений на начальных позициях, когда после начала строки сразу идёт число без проблем.

Шаблон гарантирует, что мы всегда имеем дело с корректным арифметическим выражением, так как он требует, чтобы выражение строилось строго по правилам: числа без ведущих нулей, чередующиеся с операциями, и ни в коем случае не две операции/

Можно написать решение с использованием метода левой L и правой границы подстроки R и проверки вырезки между позициями L и R. Это менее элегантный, но тоже рабочий алгоритм. Важно проверить все те же условия, что в первом решении заложены в шаблон.

В конце проверяем не заканчивается ли выражение арифметической операцией, если да, то сокращаем выражение на этот символ.

```
s = open('307_24.txt').readline()
```

```

def valid( p ):
    if p[0]in "4567":
        if p.count("--")==0 and p.count("-*")==0 and p.count("*-")==0 and p.count("**")==0:
            if p.count("-04")==0 and p.count("-05")==0 and p.count("-06")==0 and p.count("-07")==0 and p.count("-00")==0:
                if p.count("*04")==0 and p.count("*05")==0 and p.count("*06")==0 and p.count("*07")==0 and
p.count("*00")==0:
                    return True
        return False

N = len(s)
maxLen = 0

for L in range(N):
    for R in range(L+maxLen+1,N+1):
        sub = s[L:R]
        if valid( sub ):
            maxLen = R - L
            sMax = sub
        else:
            break

if sMax[-1] in "*-":
    maxLen -=1
    sMax = sMax[:-1]
print( maxLen)
print( sMax )

```

В примере печатается и сама строка (для проверки), в качестве ответа в задании требуется вывести только значение длины – maxLen.

Анализ ошибок

Прежде всего, нужно понимать, что это задание на программирование и нужно владеть языком программирования и обладать развитым алгоритмическим мышлением. Типичные ошибки обучающихся при составлении программы:

1. ***Игнорирование ведущих нулей*** – допускают числа вида 05, 007 (в условии запрещено).
2. ***Разрешение пустого выражения или выражения без чисел*** – например, пустая строка или только операции.
3. ***Неучёт отдельного нуля*** – считают, что 0 не может быть числом, хотя это допустимо.
4. ***Разрешение двух операций подряд*** – не проверяют, что между операциями обязательно должно быть число.
5. ***Разрешение операции в конце или начале*** – выражение не должно начинаться или заканчиваться на - или *.
6. ***Поиск только непересекающихся подстрок*** – используют `re.findall` без `lookahead` и пропускают перекрывающиеся выражения.
7. ***Неправильная обработка знака «-»*** – путают его с разделителем или считают частью числа.
8. ***Жадный перебор без проверки корректности*** – просто ищут самую длинную строку без операций подряд.
9. ***Не учитывают, что число может состоять только из цифр 4,5,6,7,0***.

Несформированные метапредметные умения

Низкий процент выполнения задания (0,96% в группе «61–80», 15,9% в группе «81–100», разрыв 14,94 п.п.) связан с несформированностью:

базовых исследовательских действий (1.2.4, 1.2.5, 1.2.7) – неумение организовать обработку символьных строк, отсутствие навыка использования регулярных выражений или ручного парсинга, неумение формализовать правила построения корректного выражения;

регулятивных УУД (Самоорганизация) (3.1) – неумение составить план решения, определить последовательность шагов, выбрать эффективный инструмент (регулярные выражения или ручной перебор);

регулятивных УУД (Самоконтроль) (3.2) – отсутствие тестирования на малых примерах, неумение проверить корректность выражения на граничных случаях.

Способы избежать таких ошибок:

Задание 24 проверяет не базовые, а продвинутые навыки владения программированием и посильна только хорошо

подготовленным школьникам, изучавшим углубленный курс информатики. Поэтому, тренировать решение таких задач в классе стоит с сильными обучающимися. Важно при освоении навыков решения таких задач показать обучающимся более эффективные способы разработки алгоритма, иначе программа может выполняться долго: медот динамического программирования (применение левых и правых границ); в некоторых случаях хорошо работает разбиение строки (в Python это метод `split()`). Важно учить выделять все условия в формулировке задания и тестировать промежуточные результаты.

Следует придерживаться правил:

- 1) Сначала формализуй правила на бумаге: что такое число, что такое выражение.
- 2) Выбери инструмент: регулярки (быстро, надёжно) или ручной парсинг (наглядно, контролируемо).
- 3) Для регулярок используй (`?=...`), чтобы не потерять перекрывающиеся подстроки.
- 4) При ручном переборе проверяй структуру слева направо, не пытайся исправлять ошибки постфактум.

Чего избегать

- 1) Не используй `\d` – в файле только цифры 0,4,5,6,7.
- 2) Не проверяй только операции подряд – проверяй всю структуру целиком.
- 3) Не ищи только непересекающиеся строки – выражение может быть внутри другого.
- 4) Не разрешай выражения из одного числа – в условии "непрерывная последовательность", но обычно ищут с операциями; уточни по примерам.

Как проверить себя

- 1) Создай тестовую строку: "04*5", "4-05", "4--5", "4*", "0", "45*67".
- 2) Прогони своё решение и убедись, что некорректные не проходят.
- 3) Проверь, что находится самое длинное выражение, а не просто валидное.

Для другой формулировки, конечно требуется проверки других условий и алгоритмы будут абсолютно отличаться, но в целом методы их построения, приемы и пошаговый контроль выполняются аналогично.

Задание 25

Тема: Обработка целых чисел. Проверка делимости.

Уровень сложности: высокий.

Рекомендуемое время выполнения: 20 минут.

Умение создавать собственные программы (10–20 строк) для обработки целочисленной информации.

Проверяемые элементы содержания: Алгоритмы обработки натуральных чисел, записанных в позиционных системах счисления:

разбиение записи числа на отдельные цифры, нахождение суммы и произведения цифр, нахождение максимальной (минимальной) цифры. Представление числа в виде набора простых сомножителей. Алгоритм быстрого возведения в степень. Поиск простых чисел в заданном диапазоне с помощью алгоритма «решето Эратосфена».

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения сред него арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива; умение использовать в программах данные различных типов с учётом ограничений на диапазон их возможных значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; умение использовать средства отладки программ в среде программирования

Проверяемые метапредметные результаты: базовые исследовательские действия, самоорганизация, самоконтроль.

Что нужно знать:

задачи этого типа можно решать в электронных таблицах или с помощью собственной программы, как правило, написать правильную программу значительно проще;

пусть необходимо перебрать все целые числа на отрезке $[a; b]$ и подсчитать, для скольких из них выполняется некоторое условие, общая структура цикла перебора записывается так (Python):

```
count = 0
for n in range(a, b+1):
    if условие выполнено:
        count += 1
```

```
print( count )
```

Pascal:

```
count := 0;
for n:=a to b do
  if условие выполнено then
    count := count + 1;
writeln(count);
```

C++:

```
int count = 0;
for(int n = a; n <= b; n++)
  if( условие выполнено )
    count += 1;
std::cout << count;
```

проверку условия удобно оформить в виде функции, возвращающей логическое значение (True/False), но можно этого и не делать;

проверить делимость числа n на число d можно с помощью операции взятия остатка от деления n на x : если остаток равен 0, число n делится на x нацело;

проверка делимости на языке Python выглядит так:

```
if n % d == 0:
    print("Делится")
else: print("Не делится")
```

тоже самое на Pascal

```
if n mod d = 0 then
  writeln('Делится')
else writeln('Не делится')
```

```

то же самое на C++
if( n % d == 0 )
    std::cout << "Делится";
else std::cout << "Не делится";

```

Количество делителей

для определения числа делителей натурального числа n можно использовать цикл, в котором перебираются все возможные делители d от 1 до n , при обнаружении делителя увеличивается счётчик делителей:

```

count = 0
for d in range(1, n+1):
    if n % d == 0:
        count += 1
print( count ) # вывести количество делителей

```

тоже самое на Pascal

```

count := 0;
for d:=1 to n do
    if n mod d = 0 then
        count := count + 1;
writeln( count );

```

то же самое на C++

```

int count = 0;
for(int d = 1; d <= n; d++)
    if( n % d == 0 ) count ++;
std::cout << count; // вывести количество делителей

```

если требуется определить не только количество делителей, но и сами делители, нужно сохранять их в массиве; в языке Python удобно использовать динамический массив: сначала он пуст, а при обнаружении очередного делителя этот делитель добавляется в массив:

```

divs = []

```

```
for d in range(1,n+1): # перебор всех возможных делителей
```

```
    if n % d == 0:      # если нашли делитель d
```

```
        divs.append(d)  # то добавили его в массив
```

в языках Pascal и C++ проще обойтись без динамического массива; здесь есть два варианта:

1) выделить массив достаточного размера для хранения всех делителей; например, количество делителей числа n явно не превышает n ;

2) хранить только нужное количество делителей, например, если нас интересуют числа, имеющие 4 делителя, достаточно выделить массив из 4-х элементов, а остальные делители в массив не записывать;

перебор делителей можно оптимизировать, учитывая, что наименьший из пары делителей, таких что $a \times b = n$, не превышает квадратного корня из n ; нужно только аккуратно обработать случай, когда число n представляет собой квадрат другого целого числа;

отметим, что для чисел, которые предлагаются в вариантах заданий, такая оптимизация не обязательна; более того, усложнение программы может привести к дополнительным ошибкам...

Простые числа:

простое число n делится только на 1 и само на себя, причём единица не считается простым числом; таким образом, любое простое число имеет только два делителя

для определения простоты числа можно считать общее количество его делителей; если их ровно два, то число простое, если не два – не простое:

```
nDel = 0          # количество делителей числа
```

```
for d in range(1, n+1): # все возможные делители
```

```
    if n % d == 0:
```

```
        nDel += 1      # нашли ещё делитель
```

```
if nDel == 2:
```

```
    print( "Число простое" )
```

```
else:
```

```
    print( "Число составное" )
```

работу программы можно ускорить: если уже найдено больше двух делителей, то число не простое и можно досрочно закончить работу цикла с помощью оператора break:

```

nDel = 0          # количество делителей числа
for d in range(1, n+1): # все возможные делители
    if n % d == 0:
        nDel += 1    # нашли ещё делитель
        if nDel > 2:  # уже не простое число
            break     # досрочный выход из цикла
if nDel == 2:
    print( "Число простое" )
else:
    print( "Число составное" )

```

другой вариант – считать количество делителей числа на отрезке $[2; n-1]$; как только хотя бы один такой делитель будет найден, можно завершить цикл, потому что число явно не простое:

```

nDel = 0          # количество делителей на отрезке [2; n-1]
for d in range(2, n):
    if n % d == 0:
        nDel += 1    # нашли делитель
        break       # досрочный выход из цикла
if nDel == 0:
    print( "Число простое" )
else:
    print( "Число составное" )

```

можно сделать то же самое с помощью логической переменной:

```

prime = True      # сначала считаем число простым
for d in range(2, n):
    if n % d == 0:
        prime = False # уже не простое
        break         # досрочный выход из цикла
if prime:
    print( "Число простое" )

```

```

else:
    print( "Число составное" )
тоже самое на Pascal
prime := True;           { сначала считаем число простым }
for d:=2 to n-1 do
    if n mod d = 0 then begin
        prime := False; { уже не простое }
        break           { досрочный выход из цикла }
    end;
if prime then
    writeln( 'Число простое' )
else
    writeln( 'Число составное' );

```

```

то же самое на C++
bool prime = true;           // сначала считаем число простым
for( int d = 2; d <= n-1; d++ )
    if( n % d == 0 ) {
        prime = false; // уже не простое
        break;         // досрочный выход из цикла
    }
if( count == 2 )
    std::cout << "Число простое";
else
    std::cout << "Число составное";

```

в этом задании обычно предлагаются большие числа, поэтому проверка делимости на все числа от 2 до $n-1$ выполняется очень долго, и на устаревших компьютерах время работы приведённого выше алгоритма может быть слишком велико

программу можно оптимизировать, если вспомнить, что наименьший из пары делителей, таких что $a \times b = n$, не

превышает квадратного корня из n ; поэтому можно закончить перебор значением, округлив его до ближайшего целого числа; если на отрезке $[2; \sqrt{n}]$ не найден ни один делитель, их нет и на отрезке $[\sqrt{n} + 1, n - 1]$

следовательно, можно существенно ускорить перебор, изменив конечное значение переменной цикла:

for d in range(2, round(sqrt(n))+1):

на языке Pascal:

for d:=2 to round(sqrt(n)) do

на языке C++:

for(int d = 2; d <= round(sqrt(n)); d++)

Пример формулировки задания (используется открытый вариант №307)

Напишите программу, которая перебирает целые числа, бóльшие 2 018 974 447, в порядке возрастания и ищет среди них числа, представленные в виде произведения ровно двух простых множителей, не обязательно различных, каждый из которых ровно один раз содержит в своей записи 43 (43 – идущие подряд друг за другом в указанном порядке цифры 4 и 3). В ответе в первом столбце таблицы запишите первые 5 найденных чисел в порядке возрастания, а во втором столбце – для каждого из них соответствующий наименьший найденный множитель. Количество строк в таблице для ответа избыточно.

Решение (Python)

```
def f(x):
```

```
    for i in range(2, int(x**0.5)+1):
```

```
        if x%i==0:
```

```
            return [i]+f(x//i)
```

```
    return [x]
```

```
count = 0
```

```
for x in range(2018974448, 2100000000):
```

```
    d = f(x)
```

```
    if len(d) == 2:
```

```
        if str(d[0]).count('43') == 1 and str(d[1]).count('43') == 1:
```

```

print(x, min(d))
count += 1
if count == 5:
    break

```

Это одно из самых элегантных решений и в целом требует не много времени для выполнения. Главное достоинство программы – поиск с помощью функции $f(x)$ разложения числа на простые множители и если в разложении ровно два простых множителя, остается проверить есть ли в их записи ровно 1 раз «43».

Можно использовать и две более стандартные подпрограммы для основного алгоритма: поиск делителей и проверки простоты делителя. Это будет чуть менее эффективный алгоритм с большей длиной кода, но тоже рабочий.

Анализ ошибок

Типичные ошибки обучающихся:

1. Недопонимание условия задачи:

ошибочное понимание требований: некоторые обучающийся считают, что число должно быть представлено произведением любого количества простых множителей, хотя в условии строго оговорено, что множителей должно быть ровно два.

неточность в понимании термина "простое число": некоторые ошибочно полагают, что любое число делится на единицу и само себя, но не осознают, что единица не считается простым числом.

2. Проблемы с обработкой случаев:

использование одного критерия: многие решают задачу, полагая, что нужно просто находить любые простые множители, а потом вручную подбирать те, что содержат цифры 43, забывая, что оба множителя должны содержать ровно один раз 43.

3. Логические ошибки:

искажённое представление о составе числа: некоторые пытаются использовать формулу разложения на множители (включая степени), забывая, что задача требует именно произведения двух простых чисел.

игнорирование порядка: многие обучающийся ищут пару множителей и не задумываются о порядке следования множителей, что увеличивает вероятность пропуска правильного варианта.

4. Ошибки программирования:

программирование процесса факторизации: ошибка возникает при неправильном определении способа проверки

деления числа на потенциальные множители.

выбор подходящего метода поиска простых чисел: использование неэффективных методов поиска простых чисел замедляет выполнение программы, что усложняет процесс тестирования больших чисел.

работа с большим количеством данных: некоторые выбирают неэффективные способы хранения промежуточных результатов, что сильно замедляет скорость выполнения программы.

5. Технические аспекты:

испытание большого объема чисел: многие студенты забывают установить разумные верхние границы для поиска чисел, что затрудняет тестирование больших наборов данных.

преждевременное завершение программы: ошибки могут возникать, если цикл завершён преждевременно или выполняется дольше необходимого срока.

Несформированные метапредметные умения

Низкий процент выполнения задания (8,27% в группе «61–80», 60,67% в группе «81–100», разрыв 52,40 п.п.) связан с несформированностью:

базовых исследовательских действий (1.2.4, 1.2.5, 1.2.7) – неумение организовать обработку числовых последовательностей, отсутствие навыка реализации алгоритмов поиска и сортировки, неумение использовать структуры данных;

регулятивных УУД (Самоорганизация) (3.1) – неумение составить план решения, определить последовательность шагов, выбрать эффективный алгоритм;

регулятивных УУД (Самоконтроль) (3.2) – отсутствие тестирования на малых примерах, неумение проверить корректность алгоритма на граничных случаях.

Примеры ошибок: неправильная реализация алгоритма поиска; ошибки в сортировке; неверное использование структур данных; проблемы с индексацией; отсутствие проверки границ; неправильная инициализация переменных; недопонимание условия задачи (например, что множителей должно быть ровно два).

Способы избежать таких ошибок:

Задание 25 так же, как и задание 24 проверяет не базовые, а продвинутые навыки владения программированием и посильна только хорошо подготовленным школьникам, изучавшим углубленный курс информатики. Поэтому, тренировать решение таких задач в классе стоит с сильными обучающимися. Однако, поиск делителей числа и

нахождение простых чисел, это базовые алгоритмы, которые можно разбирать при изучении циклов, и овладение ими можно реализовать со школьниками в начале изучения основ программирования.

Важно при освоении навыков решения таких задач отрабатывать с обучающимися:

эффективные приемы поиска делителей и простых чисел, разложения на простые множители;

тщательное тестирование алгоритма на небольших примерах;

эффективные методы факторизации и поиска простых чисел, чтобы минимизировать временные затраты;

организацию решения таким образом, чтобы легко было проследить ход рассуждений и выявить возможные ошибки.

Задание 26

Тема: Обработка массива целых чисел из файла. Сортировка.

Уровень сложности: высокий.

Рекомендуемое время выполнения: 35 минут.

Умение обрабатывать целочисленную информацию с использованием сортировки.

Проверяемые элементы содержания: Массивы и последовательности чисел. Вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения, среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию). Линейный поиск заданного значения в массиве. Алгоритмы работы с элементами массива с однократным просмотром массива. Сортировка одномерного массива. Простые методы сортировки (метод пузырька, метод выбора, сортировка вставками). Сортировка слиянием. Быстрая сортировка массива (алгоритм QuickSort). Двоичный поиск в отсортированном массиве.

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения, среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива; умение использовать в программах данные различных типов с учётом ограничений на диапазон их возможных значений, применять при решении задач структуры данных (списки,

словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; умение использовать средства отладки программ в среде программирования.

Проверяемые метапредметные результаты: базовые исследовательские действия, самоорганизация, самоконтроль.

Что нужно знать (К.Ю. Поляков):

Чтение данных из файла

в языке Python для чтения данных удобно использовать менеджер контекста (**with ... as**), который открывает файл и закрывает его; например, код

```
with open("26.txt") as Fin: # программа и файл в одной папке  
... # какие-то операции с файлом  
# при завершении работы менеджера контекста  
# файл автоматически закрывается
```

равносителен такому

```
Fin = open("26.txt") # открытие файла  
... # какие-то операции с файлом  
Fin.close() # закрытие файла
```

– если в текущей строке файла находится целое число, то прочитать его в переменную **x** можно так:

```
x = int( Fin.readline() )
```

если в строке записаны два числа, после чтения (**Fin.readline()**) строку нужно разбить на отдельные части по пробелам между числами (каждая часть – символьная запись числа) и затем каждую часть преобразовать в целое число; например, чтение двух чисел:

```
s = Fin.readline()  
symData = s.split()  
x, y = map( int, symData )  
или в компактной форме  
x, y = map( int, Fin.readline().split() )
```

в языке PascalABC.NET для чтения данных проще всего просто перенаправить входной поток на файл:

```
Assign( input, '26.txt' );
```

после этого можно использовать операторы **read** и **readln**, так же, как при вводе с клавиатуры в языке C++ можно читать данные с помощью входного потока (**fstream**):

```
#include <fstream>
```

```
...
```

```
ifstream Fin("26.txt");
```

```
Fin >> x;
```

```
Fin >> y >> z;
```

Хранение массива данных

в языке Python для хранения массива данных используется список; следующая программа показывает чтение массива данных размера N в список **data** из файла «26.txt» (данные записаны в столбик, по одному числу в строке):

```
data = [0]*N
```

```
with open("26.txt") as Fin:
```

```
    for i in range(N):
```

```
        data[i] = int( Fin.readline() )
```

или с помощью генератора списка

```
with open("26.txt") as Fin:
```

```
    data = [ int( Fin.readline() )
```

```
        for i in range(N) ]
```

в языке PascalABC.NET используем динамический массив; когда станет известен его размер, выделяем место в памяти и читаем из входного потока:

```
var data: array of integer;
```

```
SetLength( data, N );
```

```
for var i:=0 to N-1 do
```

```
    read( data[i] );
```

в языке C++ аналогично используется коллекция **vector**:

```
#include <vector>
```

```
...
```

```
vector <int> data(N);
```

```
for( int i = 0; i < N; i++ )
```

```
Fin >> data[i];
```

Сортировка массива

Для сортировки имеет смысл использовать встроенные функции языков программирования. Категорически НЕ рекомендуется писать собственные реализации алгоритмов сортировки.

В языке Python для сортировки массива (списка) «на месте» вызывается метод **sort**:

```
data.sort()
```

при этом числа сортируются по возрастанию. Для сортировки по убыванию в вызов метода добавляем именованный аргумент **reverse** со значением **True**:

```
data.sort( reverse = True )
```

Для сортировки по другому критерию (например, по последней цифре числа) добавляют именованный аргумент **key**, который указывает на функцию, вычисляющую нужное значение, например:

```
def lastDigit( n ):
```

```
    return n % 10
```

```
... # заполнение массива data
```

```
data.sort( key = lastDigit )
```

Простую функцию можно не оформлять как отдельную подпрограмму, а записать как неименованную функцию (лямбда-функцию) :

```
data.sort( key = lambda x: x % 10 )
```

Иногда данные в массиве **data** представляют собой пары или тройки чисел, объединённые в кортежи. В этом случае при стандартной сортировке сначала сравниваются первые элементы кортежей, если они равны – вторые и т.д. Чтобы задать свой порядок сортировки, нужно использовать аргумент **key** с обычной функцией или лямбда-функцией. Например,

```
data.sort( key = lambda x: (-x[1], x[0]%10) )
```

В этом примере происходит сортировка по убыванию (знак «минус») второго числа в кортеже, **x[1]**, а если вторые элементы равны - по возрастанию последней цифры первого элемента кортежа, **x[0]**.

Если нужно создать новый массив, не изменяя исходные данные, используется функция **sorted**. Её первый аргумент – массив, а остальные совпадают с аргументами метода **sort**. Например,

```
data1 = sorted( data, key = lambda x: (-x[1], x[0]%10) )
```

(Е. Джобс) В языке Python возможна сортировка строк двумерного массива. При этом сначала выполняется

сортировка по первому элементу в каждой строке, потом – по второму и т.д. Например, что при сортировке двумерного (точно работает) списка, сортировка идет по значениям второй размерности слева направо. Например, результатом вызова

```
sorted([ [3,5], [1,6], [2,8], [3,4], [5,8] ])
```

будет

```
[ [1,6], [2,8], [3,4], [3, 5], [5,8] ]
```

В языке PascalABC.NET для динамических массивов используется метод **Sort**:

```
data.Sort;
```

По умолчанию сортировка выполняется в порядке возрастания.

Для нестандартной сортировки лучше использовать метод **OrderBy**, в качестве аргумента можно указать лямбда-функцию. При этом строится новый массив. Вот пример с сортировкой по возрастанию последней цифры числа:

```
var data1 := data.OrderBy( x->x mod 10).ToArray;
```

Если нужна сортировка по убыванию, вместо **OrderBy** применяется метод **OrderByDescending**.

в языке C++ для сортировки коллекции **vector** вызывается процедура **sort** (сортировка «на месте»):

```
#include <vector>
```

```
...
```

```
vector <int> data(N);
```

```
...
```

```
sort( data.begin(), data.end() );
```

Для сортировки по убыванию третьим аргументом указывается функция **greater<int>**:

```
sort( data.begin(), data.end(), greater<int>() );
```

Нестандартную формировку можно выполнить с помощью собственной функции сравнения. Вот, например, сортировка вектора по возрастанию последней цифры:

```
bool cmpLastDigit( int i1, int i2)
```

```
{
```

```
    return (i1 % 10 < i2 % 10);
```

```
}
```

```
...
```

```
sort( data.begin(), data.end(), cmpLastDigit );
```

Можно использовать и аналогичную лямбда-функцию:

```
sort( data.begin(), data.end(),
      []( int i1, int i2) { return (i1 % 10 < i2 % 10); } );
```

Пример формулировки задания (используется открытый вариант №307)

Сервер выполняет запросы на передачу данных, при этом сведения о каждом выполненном запросе (время регистрации, идентификатор клиента и объём переданных данных) сохраняются в журнале работы, а сам запрос – в специальном разделе памяти сервера, имеющем ограниченный объём. Каждый раз, когда в специальном разделе остаётся недостаточно свободной памяти, сервер создаёт резервную копию всех накопленных там данных, после чего освобождает раздел и продолжает выполнение запросов. Напишите программу для обработки журнала работы сервера и с её помощью определите идентификатор клиентского устройства, с которого на сервер был передан наибольший суммарный объём данных, не превышающий 150 000 Кбайт, а также сумму объёмов двух наибольших резервных копий специального раздела (в Кбайт), выполненных не ранее 12:00:00.

Входные данные

Первая строка входного файла (журнала работы сервера) содержит два натуральных числа: N ($N < 1\,000\,000$) – количество строк в журнале и K ($K < 1\,000\,000$) – вместимость специального раздела памяти сервера в Кбайт. Каждая из следующих N строк содержит информацию об одном выполненном запросе: время регистрации запроса в формате ЧЧ:ММ:СС (часы, минуты, секунды) и два натуральных числа: C ($C < 1\,000\,000$) – идентификатор клиентского устройства и S ($S < K$) – объём данных запроса в Кбайт.

Выходные данные

В ответе запишите два числа: сначала идентификатор устройства, с которого на сервер был передан наибольший суммарный объём данных, не превышающий 150 000 Кбайт, а затем сумму объёмов двух наибольших резервных копий специального раздела (в Кбайт), выполненных не ранее 12:00:00.

Типовой пример организации данных во входном файле

8 140000

01:01:01 101 20000

03:03:03 202 110000

05:05:05 101 90000

07:07:07 303 62000

10:10:10 101 48000
 15:15:15 202 12000
 21:21:21 303 120000
 23:23:23 404 134000

При таких исходных данных резервное копирование специального раздела выполняется четыре раза: в 05:05:05 (в объёме 130 000 Кбайт), в 07:07:07 (в объёме 90 000 Кбайт), в 21:21:21 (в объёме 122 000 Кбайт) и в 23:23:23 (в объёме 120 000 Кбайт). Всего на сервер передано 596 000 Кбайт данных: 158 000, 122 000, 182 000 и 134 000 Кбайт от клиентов с идентификаторами 101, 202, 303 и 404 соответственно. Ответ для приведённого примера: 404 242 000.

Типовой пример имеет иллюстративный характер. Для выполнения задания используйте данные из прилагаемого файла.

Решение (Python)

```
f = open('307_26.txt')
n, k = map(int, f.readline().split())
d = {}
mem = 0
a = []

for _ in range(n):
    t, id, s = f.readline().split()
    id, s = int(id), int(s)

    if id not in d:
        d[id] = 0
    d[id] += s

    if mem + s > k:
        if t >= '12:00:00':
            a.append(mem)
```

```

        mem = s
    else:
        mem += s

mx = 0
for i in d:
    if d[i] <= 150000 and d[i] > mx :
        mx = d[i]
        client = i

a.sort(reverse=True)
print(client, a[0] + a[1])

```

Анализ ошибок

В данной группе обучающихся к заданию 26 приступило всего 18,85%, при этом 17,12% не справились с заданием, только на один вопрос верно ответили 1,54%, на 2 вопроса правильно ответили 0,19%. Типичные ошибки обучающихся:

1. **Логика резервирования:** При $mem + s > k$ ошибочно обнуляют mem вместо $mem = s$. Запрос, не поместившийся в раздел, сам становится началом нового раздела.
2. **Время:** Сравнивают строки в неправильном формате, переводят в секунды (избыточно) или игнорируют условие "не ранее 12:00:00" при добавлении копий в список.
3. **Сумма клиента:** Суммируют только неудавшиеся запросы (внутри условия $if\ mem + s > k$), хотя нужно суммировать все запросы клиента независимо от резервирования.
4. **Момент копирования:** Добавляют в список $mem + s$ вместо mem . Резервная копия создаётся до добавления нового запроса, а не после.
5. **Поиск клиента:** Игнорируют условие ≤ 150000 , находят просто максимум, либо неправильно инициализируют mx (например, $mx = -1$ или не сбрасывают при отсутствии подходящих).
6. **Список копий:** Не проверяют наличие хотя бы двух элементов ($len(a) \geq 2$), сортируют по возрастанию и берут $a[-1] + a[-2]$ (работает, но менее читаемо).
7. **Чтение данных:** Не преобразуют id и s в `int` или используют неэффективные методы чтения

(например, `readlines()` для больших файлов).

8. **Типы данных:** Используют `float` для больших объёмов (потеря точности) или сравнивают `id` как строки (лексикографическое сравнение).

Ключевое: правильное решение суммирует все запросы клиента, при превышении `k` сохраняет `tem` (если время $\geq 12:00:00$) и присваивает `tem = s`, а затем находит клиента с максимальной суммой ≤ 150000 и сумму двух наибольших копий.

Несформированные метапредметные умения

Низкий процент выполнения задания (0,96% в группе «61–80», 24,06% в группе «81–100», разрыв 23,10 п.п.) связан с несформированностью:

базовых исследовательских действий (1.2.4, 1.2.5, 1.2.7) – неумение организовать обработку целочисленной информации с использованием сортировки, отсутствие навыка реализации сложных алгоритмов обработки данных;

регулятивных УУД (Самоорганизация) (3.1) – неумение составить план решения, определить последовательность шагов, выбрать эффективный алгоритм;

регулятивных УУД (Самоконтроль) (3.2) – отсутствие тестирования на малых примерах, неумение проверить корректность алгоритма на граничных случаях;

работы с информацией (1.3.2., 1.3.4.) – неумение правильно прочитать данные из файла, ошибки при преобразовании типов данных.

Способы избежать таких ошибок:

Задание 26 проверяет не базовые, а продвинутые навыки владения программированием и посильна только хорошо подготовленным школьникам, изучавшим углубленный курс информатики. Поэтому, тренировать решение таких задач в классе стоит с сильными обучающимися.

При подготовке школьников к решению таких задач, важно акцентировать внимание на том, что важна сортировка для лучшего поиска, научить использовать чёткий критерий сортировки по времени окончания с использованием `lambda` переменной.

Для правильного ответа на второй вопрос нужно выбирать два самых больших значения, что важно, полученные не ранее 12:00.

Задание наглядно выполняется с помощью электронных таблиц. Возможно, обучающиеся, не очень уверенно

программирующие, лучше поймут такой способ решения, хотя, учитывая низкий процент выполнимости задания 9, это не однозначно.

Задание 27

Тема: Анализ данных. Кластеризация.

Уровень сложности: высокий.

Рекомендуемое время выполнения: 40 минут.

Умение выполнять последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и построение модели, преобразование данных, визуализация данных, интерпретация результатов.

Проверяемые элементы содержания: Анализ данных. Основные задачи анализа данных: прогнозирование, классификация, кластеризация, анализ отклонений. Последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и/или построение модели, преобразование данных, визуализация данных, интерпретация результатов. Программные средства и интернет-сервисы для обработки и представления данных. Большие данные. Машинное обучение.

Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС 2021 г.: Умение классифицировать основные задачи анализа данных (прогнозирование, классификация, кластеризация, анализ отклонений); понимать последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и/или построение модели, преобразование данных, визуализация данных, интерпретация результатов.

Проверяемые метапредметные результаты: базовые исследовательские действия.

Что нужно знать:

Чтение данных из файла

в языке Python для чтения данных удобно использовать менеджер контекста (**with ... as**), который открывает файл и закрывает его; например, код

```
with open("26.txt") as Fin: # программа и файл в одной папке  
... # какие-то операции с файлом  
# при завершении работы менеджера контекста  
# файл автоматически закрывается
```

равносилен такому

```
Fin = open("26.txt") # открытие файла
```

```
... # какие-то операции с файлом
```

```
Fin.close() # закрытие файла
```

если в текущей строке файла находится целое число, то прочитать его в переменную **x** можно так:

```
x = int( Fin.readline() )
```

если в строке записаны два числа, после чтения (**Fin.readline()**) строку нужно разбить на отдельные части по пробелам между числами (каждая часть – символьная запись числа) и затем каждую часть преобразовать в целое число; например, чтение двух чисел:

```
s = Fin.readline()
```

```
symData = s.split()
```

```
x, y = map( int, symData )
```

или в компактной форме

```
x, y = map( int, Fin.readline().split() )
```

в языке PascalABC.NET для чтения данных проще всего просто перенаправить входной поток на файл:

```
Assign( input, '26.txt' );
```

после этого можно использовать операторы **read** и **readln**, так же, как при вводе с клавиатуры

в языке C++ можно читать данные с помощью входного потока (**fstream**):

```
#include <fstream>
```

```
...
```

```
ifstream Fin("26.txt");
```

```
Fin >> x;
```

```
Fin >> y >> z;
```

Принципы кластеризации.

Способы визуализации (диаграммы в электронных таблицах, работа с графикой в программировании).

Пример формулировки задания (используется открытый вариант №307)

Фрагмент звёздного неба спроецирован на плоскость с декартовой системой координат. Учёный решил провести кластеризацию полученных точек, являющихся изображениями звёзд, то есть разбить их множество на N непересекающихся непустых подмножеств (кластеров), таких, что точки каждого подмножества лежат внутри прямоугольника со сторонами длиной H и W , причём эти прямоугольники между собой не пересекаются. Стороны прямоугольников не обязательно параллельны координатным осям. Гарантируется, что такое разбиение существует и единственно для заданных размеров прямоугольников.

Будем называть центром кластера точку (звезду) этого кластера, сумма расстояний от которой до всех остальных его точек минимальна. Для каждого кластера гарантируется единственность его центра. Расстояние между двумя точками на плоскости $A(x_1, y_1)$ и $B(x_2, y_2)$ вычисляется по формуле:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Каждая звезда помимо координат на плоской карте характеризуется своим спектральным классом и классом светимости. Спектральный класс определяет цвет (который связан с температурой звезды) следующим образом.

Обозначение спектрального класса (латинская буква)	О	В	А	F	G	К	М
Цвет звезды	Голубой	Бело-голубой	Белый	Жёлто-белый	Жёлтый	Оранжевый	Красный

Каждый из спектральных классов, в свою очередь, делится на подклассы от 0 до 9 в порядке уменьшения температуры. Обозначение подкласса ставится после обозначения спектрального класса (например, B2). Класс светимости звезды обозначим римскими цифрами от I до VII.

Обозначение класса светимости	I	II	III	IV	V	VI	VII
Светимость	Сверх-гигант	Яркий гигант	Гигант	Суб-гигант	Карлик	Суб-карлик	Белый карлик

В файле А хранится информация о точках двух кластеров, где $N = 6,0$ и $W = 5,5$ для каждого кластера. В каждой строке сначала записана информация о расположении на карте одной звезды: координата x и координата y . Далее в той же строке для звезд классов светимости I–VI указываются спектральный класс, подкласс и класс светимости. Обозначения классов ничем не разделяются. Для звезд класса светимости VII (Белый карлик) обозначения спектрального класса и подкласса в файле не указываются. Известно, что количество точек не превышает 2000.

В файле Б хранятся координаты точек трех кластеров, где $N = 6,0$ и $W = 5,5$ для каждого кластера. Известно, что количество точек не превышает 10 000. Структура хранения информации в файле Б аналогична структуре в файле А.

Для файла А определите координаты центра каждого кластера, затем найдите два числа: A_x и A_y – абсциссу и ординату бело-голубого яркого гиганта, ближайшего к центру кластера, который содержит наибольшее количество точек.

Для файла Б определите координаты центра каждого кластера, затем найдите два числа: B_1 – расстояние между центрами кластеров с наименьшим и наибольшим количеством желтых карликов и B_2 – наибольшее расстояние между оранжевыми карликами одного кластера.

В ответе запишите четыре числа: в первой строке – сначала целую часть абсолютной величины произведения $A_x \times 10\,000$, затем целую часть абсолютной величины произведения $A_y \times 10\,000$; во второй строке – сначала целую часть произведения $B_1 \times 10\,000$, затем целую часть произведения $B_2 \times 10\,000$.

Пример организации данных в одном из исходных файлов для случая четырех звезд

5,01788	8,32466	G2V
4,289251	6,955186	VII
4,619358	5,524697	B7V
6,91934	20,425391	G2V

Внимание! Пример приведен в иллюстративных целях для произвольных значений, не имеющих отношения к заданию. Для выполнения задания используйте данные из прилагаемых файлов.

Решение (Python)

```
data=[]
for s in open("307_27_B.txt"): #"307_27_B.txt"
    x,y,z=s.split()
    x=float(x.replace(",","."))
    y=float(y.replace(",","."))
```

```

data.append([x,y,z])

#находим кластеры точек, для которых соседями являются точкам на расстоянии <d
from math import dist
clusters=[]
d=1
while data:
    cl=[data.pop()]
    for p in cl:
        sosed=[p1 for p1 in data if dist((p[0],p[1]),(p1[0],p1[1]))<d]
        for p1 in sosed:
            cl.append(p1)
            data.remove(p1)
    clusters.append(cl)
#выводим количество точек в каждом полученном кластере
len_cl=[len(cl) for cl in clusters]
print(len_cl)

#С помощью модуля turtle визуализируем полученную картину
#используя для точек разных кластеров разные случайные цвета
from turtle import *
from random import *
tracer(0)
up()
for cl in clusters:
    color=random(), random(), random()
    for x,y,z in cl:
        goto(x*15, y*15-400)
        dot(2.5, color)

```

update()

#Функция для определения центров кластеров, как и в задании

```
def centr(cl):
```

```
    m=[]
```

```
    for p in cl:
```

```
        s = 0
```

```
        for p1 in cl:
```

```
            s+=dist((p[0],p[1]),(p1[0],p1[1]))
```

```
        m.append([s,p])
```

```
    return min(m)[1]
```

#Центры

```
cent = [centr(cl) for cl in clusters]
```

```
print(cent)
```

```
«»"
```

#центр наибольшего кластера

```
mx_len=0
```

```
for i in range(len(clusters)):
```

```
    if len_cl[i] > mx_len:
```

```
        mx_len = len_cl[i]
```

```
        c=cent[i]
```

```
print(c)
```

#Ответ к А

```
Ax = 0
```

```
Ay = 0
```

```
wb_gig = ["B0II", "B1II", "B2II", "B3II", "B4II", "B5II", "B6II", "B7II", "B8II", "B9II"]
```

```
min_dist=10000
```

```
for cl in clusters:
```

```

for p in cl:
    if p[2] in wb_gig:
        k=dist((p[0],p[1]),(c[0],c[1]))
        if k < min_dist:
            min_dist = k
            Ax = p[0]
            Ay = p[1]
print(int(abs(Ax * 10000)),int(abs(Ay * 10000)))
«»"
#Ответ к B
G_V = []
for cl in clusters:
    cnt = 0
    for p in cl:
        if p[2][0] == "G" and p[2][-1] == "V":
            cnt += 1
    G_V +=[cnt]
#B1
print(G_V)
p=cent[0]
c=cent[1]
B1=dist((p[0],p[1]),(c[0],c[1]))

#B2
K_V = []
for cl in clusters:
    mx_dist=0
    for p in cl:
        if p[2][0] == "K" and p[2][-1] == "V":

```

```

for p_ in cl:
    if p_[2][0] == "K" and p_[2][-1] == "V":
        s=dist((p[0],p[1]),(p_[0],p_[1]))
        if s>mx_dist:
            mx_dist = s

K_V += [mx_dist]
B2 = max(K_V)
print(int(abs(B1 * 10000)),int(abs(B2 * 10000)))

```

Анализ ошибок

Типичные ошибки обучающихся:

1. Неправильная кластеризация

Некоторые обучающийся выбирают способ кластеризации «на глаз», используя точечные диаграммы в электронных таблицах и неверно относят к кластерам отдельные точки. Особенно эта ошибка актуальна для файла В при отделении аномальных точек. При отделении кластеров с помощью программирования важно правильно построить модели (уравнения разделяющих прямых, расстояние между соседями и пр.)

2. Неверное нахождение центров кластеров

Эта ошибка встречается реже, но бывают при подсчете арифметические ошибки (неверно построена математическая модель) или перебор не всех точек кластера.

3. Неверный подсчет результата

Школьники часто для файла В выводят ответ, как и для файла А, тогда как для файла В нужно делать другие вычисления. В этом случае виновата невнимательность при прочтении задания.

Бывает, что участники экзамена забывают при выводе ответа умножать результат на 10000 и/или находить целую часть.

4. Нехватка времени на выполнение задания

Несформированные метапредметные умения

Низкий процент выполнения задания (5,77% в группе «61–80», 53,77% в группе «81–100», разрыв 48,00 п.п.)

связан с несформированностью:

базовых исследовательских действий (1.2.4, 1.2.5, 1.2.7) – неумение выполнять последовательность решения задач анализа данных, отсутствие навыка сбора, очистки, преобразования и визуализации данных;

базовых логических действий (1.1.1., 1.1.3., 1.1.4.) – неумение классифицировать основные задачи анализа данных, непонимание последовательности решения задач анализа данных;

регулятивных ууд (самоорганизация) (3.1) – неумение составить план решения, определить последовательность шагов;

регулятивных УУД (Самоконтроль) (3.2) – отсутствие проверки промежуточных результатов, неумение интерпретировать полученные результаты.

Способы избежать таких ошибок:

В целом, задание 27 нетрудное, но долгое для выполнения всех этапов решения. При подготовке школьников к ЕГЭ по информатике важно рассматривать подобные прототипы заданий; обращать внимание на детальное прочтение формулировки задания, на разные подзадания для файлов А и В, на фиксацию важных моментов; разбирать разные способы кластеризации; приучать тестировать промежуточные решения, визуализировать результаты; составлять модели для подсчета необходимых значений; правильно распределять время работы над заданиями.

○ Предположение о достигнутых и не достигнутых метапредметных результатах ФГОС

На основании анализа выполнения заданий группой участников «от 61 до 80 т.б.» можно констатировать, что:

Достигнуты на достаточном уровне:

навыки получения и обработки информации из наглядных источников (графы, схемы, таблицы) – задания № 1, № 10, № 2, № 3, № 4, № 7;

навыки работы с текстовыми редакторами и электронными таблицами – задания № 10, № 3, № 18;

умение устанавливать существенные признаки и выявлять закономерности в задачах базового и повышенного уровня сложности – задания № 2, № 3, № 4, № 7, № 12, № 13, № 14, № 16, № 18, № 20, № 22.

Не достигнуты или сформированы на недостаточном уровне:

базовые исследовательские действия – неумение выдвигать гипотезы, организовывать перебор с обоснованной границей, анализировать достоверность результатов, выбирать эффективный способ решения в заданиях повышенного и высокого уровня (№ 11, 15, 17, 23, 24, 25, 26, 27);

регулятивные УУД (самоорганизация и самоконтроль) – отсутствие навыков планирования решения, проверки промежуточных результатов, тестирования на малых примерах, отладки программы (все задания повышенного и высокого уровня);

работа с информацией – неумение формализовать задачу, создавать каскадные формулы в электронных таблицах, работать с файловыми данными (№ 9, 17, 26, 27).

Основная причина недостаточно высоких результатов в группе «от 61 до 80 т.б.» (для достижения 80–85 т.б.) связана с несформированностью метапредметных умений, необходимых для решения задач повышенного и высокого уровня сложности:

способность планировать решение и выбирать эффективный алгоритм;

навыки программирования и отладки программ – саморегуляция и контроль результатов;

умение формализовать задачу и строить математические модели.

навыки самоконтроля и проверки результатов.

Именно эти дефициты должны стать основным направлением коррекционной работы с данной группой обучающихся для достижения 80–85 тестовых баллов.

3.2.3.3. Рекомендации для учителей по совершенствованию преподавания учебного предмета группе обучающихся со средним уровнем подготовки

На основе анализа выполнения заданий ЕГЭ по информатике обучающимися со средним уровнем подготовки (группа «от 61 до 80 т.б.») можно сформулировать следующие рекомендации к их подготовке. Целевой ориентир – уверенное получение 80–85 тестовых баллов.

Рекомендация 1: Сфокусироваться на освоении «зон роста» – заданий с наибольшим потенциалом для набора баллов

Обучающиеся этой группы успешно решают задания № 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 13, 14, 16, 18, 20, 21 (базового и повышенного уровня), но имеют низкие результаты по заданиям № 9, 11, 15, 17, 22, 23, которые имеют чёткие алгоритмы решения и высокий потенциал для освоения (разрыв с группой «81–100» составляет от 26 до 53 п.п.). Эти задания могут стать основой для набора дополнительных баллов до 80–85.

Организация процесса:

Выделите на уроках время для «минуток повторения» по этим темам.

Используйте систему «накопительных карточек», где обучающийся отмечает решенные задачи по каждой теме,

переходя к следующей только после успешного выполнения 5–7 однотипных заданий из открытого банка ФИПИ.

Начинайте с заданий с чётким формульным алгоритмом (№ 9, 11, 22), затем переходите к логическим преобразованиям (№ 15), и только потом к заданиям на программирование (№ 17, 23). Это позволит постепенно наращивать сложность.

Рекомендация 2: Формировать метапредметные умения через пошаговое решение, моделирование и системный подход

Обучающиеся демонстрируют недостаточную сформированность метапредметных умений (базовых исследовательских действий, регулятивных УУД), что проявляется в неумении выдвигать гипотезы, организовывать перебор с обоснованной границей, анализировать достоверность результатов, планировать решение и проверять себя, отсюда недостаточно высокие результаты в заданиях № 9, 11, 15, 17, 22, 23 (страницы 87-106).

Приемы работы:

1. Задание №9 («Электронные таблицы»):

Научите обучающихся составлять блок-схему проверки условий на бумаге, а затем переносить её в электронную таблицу.

Используйте метод «от простого к сложному»: сначала решите задачу, где нужно найти только сумму или только максимум, а затем постепенно добавляйте новые условия.

Отрабатывайте этап формализации: учите выделять этапы построения алгоритма, моделировать, тестировать промежуточные и итоговый результат.

2. Задание №15 («Математическая логика»):

Внедрите обязательный этап «аналитики перед кодом»: перед программированием найти критические точки, где выражение может стать ложным.

Научите упрощать логику, заменяя импликацию на дизъюнкцию ($A \rightarrow B = \neg A \vee B$).

Отрабатывайте выделение «опасных» точек (где $x \in B$ и x кратно 15 одновременно).

3. Задание №17 («Обработка числовой последовательности»):

Практикуйте пошаговую отладку с `print()` – выводите промежуточные значения: минимум, остатки для каждой пары, сумму, счётчик.

Научите использовать осмысленные имена переменных (`pair`, `sum_remainders`, `min_pair_sum`), чтобы избежать путаницы.

Отрабатывайте проверку крайних случаев: если чисел меньше 2 → вернуть (0, 0); если файл пуст → обработать исключение.

4. Задание №22 («Многопроцессорные системы»):

Научите составлять таблицы занятости процессоров (по тактам) для визуализации параллельных процессов.

Отрабатывайте определение зависимых и независимых команд.

Используйте метод «ручной прокрутки» на небольших примерах, затем переходите к сложным задачам.

Задание №23 («Динамическое программирование»):

Начните с ручного подсчёта для малых значений (например, 10 → 15), чтобы обучающийся понял логику переходов.

Отрабатывайте базовые случаи рекурсии: if a == k: return 1, if a > k: return 0.

Обязательно используйте декоратор @lru_cache для кеширования результатов (при решении на Python) или организуйте итеративное заполнение массива DP.

Проверяйте условие доступности операции: $s[-2] < s[-1]$ (строго меньше).

Рекомендация 3: Развивать навыки самоконтроля и регулятивные УУД через систему проверки и рефлексии

Основная масса ошибок в группе возникает из-за отсутствия самоконтроля: обучающийся не проверяют промежуточные результаты, не тестируют решение на малых примерах, не выполняют обратную проверку, не оценивают достоверность ответа по размерности.

Методика «Двойной проверки»:

Внедрите обязательный этап «Проверки решения» после каждого задания. Научите обучающихся использовать следующие приёмы самоконтроля:

1. Обратная проверка (для заданий № 11, 14, 16):

Если нашли информационный объём – проверьте перевод единиц измерения.

Для задания №11: проверьте, что $I = K \times \log_2 N$ даёт разумное значение.

2. Проверка на граничных условиях (для заданий № 15, 17, 22, 23):

Для задания №15: проверьте найденное A на критической точке ($x=75$).

Для задания №17: проверьте, что пары являются соседними ($\text{range}(\text{len}(\text{numbers})-1)$), а не всевозможными комбинациями.

Для задания №22: проверьте, что время выполнения не может быть меньше времени самой длинной цепочки зависимых команд.

Для задания №23: протестируйте решение на малых значениях (start=10, target=15) и сравните с ручным подсчётом.

3. Проверка по размерности (для заданий № 11, 14):
Сумма цифр не может быть больше общего количества цифр $\times$ (основание-1).
Объём информации не может быть отрицательным.
4. Проверка с помощью программирования (для заданий № 17, 23):
Напишите небольшую программу для проверки ответа (особенно для заданий № 17, 23).
Сравните результат ручного решения с результатом программы.
5. Чек-лист перед сдачей решения (пример для задания №17):
Пары только соседние
Минимум всей последовательности
Остатки от обоих элементов
Сумма остатков равна минимуму
Вывод двух чисел через пробел
Обработка пустого/короткого файла

Рекомендация 4: Организовать системную работу по формированию навыков программирования и работы с данными

Обучающиеся часто не владеют достаточными навыками программирования для решения задач повышенного уровня сложности (№ 17, 23, 24–27). При этом они пытаются решать задачи вручную, хотя задания № 17 и 23 значительно проще и надёжнее решать с помощью программирования.

Системная работа:

1. Отработка базовых алгоритмов программирования:
Регулярно включайте в уроки задачи на перебор данных, поиск минимума/максимума, работу с файлами.
Отрабатывайте структуру цикла перебора: `count = 0; for n in range(a, b+1): if условие: count += 1.`
Научите проверять делимость чисел через операцию `%`.
2. Работа с файлами:
Отрабатывайте чтение данных из файла: `with open(filename, 'r') as file: numbers = list(map(int, file.read().split()))`.
Отрабатывайте обработку данных из файла для заданий № 17, 26, 27.

3. Динамическое программирование:

Начните с простых задач на подсчёт количества способов (например, подъём по лестнице), затем переходите к заданиям типа №23.

Дайте обучающимся универсальный шаблон для задач на ДП:

```
python
def solve_dp(start, target):
    dp = [0] * (target + 1)
    dp[start] = 1
    for a in range(start, target + 1):
        if dp[a] == 0:
            continue
        # переходы
    return dp[target]
```

4. Использование регулярных выражений (для задания №24):

Для сильных обучающихся отрабатывайте использование модуля re для поиска подстрок.

Научите использовать просмотр вперёд ((?=...)), чтобы не потерять перекрывающиеся подстроки.

5. Выбор инструмента:

Обсуждайте с обучающимися, какой способ решения (программирование или ЭТ) для них проще и быстрее в каждой конкретной задаче.

Для заданий №9, 17, 22, 23 показывайте как ручное решение, так и программное, чтобы обучающийся видели преимущества каждого подхода.

Акцентирование отработки умений, требуемых в решении указанных 6 тем (№ 9, 11, 15, 17, 22, 23), в совокупности с уже успешно выполняемыми заданиями, позволит учащимся из группы «от 61 до 80 т.б.» уверенно набрать 81–85 тестовых баллов. Все задания имеют стандартные, повторяющиеся алгоритмы решения и требуют сформированности навыков программирования и алгоритмического мышления.

Основные методические акценты:

Формализация и моделирование в ЭТ (№ 9).

Вычислительные алгоритмы (№ 11, 17).

Логические и математические модели (№ 15, 22).

Динамическое программирование и рекурсия (№ 23).

Рекомендуемая последовательность освоения:

Задания с чётким формульным алгоритмом (№ 9, 11, 22).

Задания на логические преобразования (№ 15).

Задания на программирование (№ 17, 23).

Это позволит постепенно наращивать сложность и формировать уверенность в решении у обучающихся со средним уровнем подготовки. Особое внимание следует уделить отработке навыков формализации задач в электронных таблицах (№ 9) и навыков программирования (№ 17, 23), так как эти задания показывают наибольший разрыв между группами и имеют высокий потенциал для набора баллов.

3.2.4. Методический анализ выполнения заданий обучающимися с высоким уровнем подготовки (в группе от 81 т.б. и выше), включая методические рекомендации для учителей

3.2.4.1. Описание предметной подготовки участников ЕГЭ с высоким уровнем подготовки, анализ типичных дефицитов в подготовке

○ *Описание достигнутых предметных результатов ФГОС: освоенных тем программы, сформированных умений и т.п. (если имеются)*

В данной группе обучаемые справились со всеми заданиями базового уровня и высокого уровня сложности: в среднем на 95% и 93% выполнимости заданий. Главная стратегическая задача обучающихся этой группы – выполнить все задания высокого уровня.

Наиболее успешно выполнено этой группой задание высокого уровня сложности: 21.

Половина обучаемых в этой группе справилась с заданиями 25 и 27, а вот с заданиями 24 и 26 не около 80% участников не справились.

Можно констатировать, что обучающимися этой группы полностью освоены все элементы содержания, соответствующие базовому уровню подготовки (подробно описаны выше) и следующие элементы содержания школьной программы по информатике и умения, соответствующие заданиям повышенного уровня:

Задание 11

Элементы содержания: Теоретические подходы к оценке количества информации. Единицы измерения количества информации. Алфавитный подход к оценке количества информации.

Умения и навыки: Умение определять информационный объём текстовых, графических и звуковых данных при заданных параметрах дискретизации. Умение определять среднюю скорость передачи данных, оценивать изменение времени передачи при изменении информационного объёма данных и характеристик канала связи.

Метапредметные результаты: *Базовые исследовательские действия* (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов).

Задание 12.

Разделы/темы школьной программы: 11 кл., п. 113.7.3. Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат.

Умения и навыки: Понимание базовых алгоритмов обработки числовой и текстовой информации (запись чисел в позиционной системе счисления, делимость целых чисел; нахождение всех простых чисел в заданном диапазоне; обработка многоразрядных целых чисел; анализ символьных строк и других), алгоритмов поиска и сортировки.

Метапредметные результаты: *Базовые логические действия* (Самостоятельно формулировать и актуализировать

проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения.). *Базовые исследовательские действия* (Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях.).

Задание 13

Элементы содержания: Сетевые протоколы. Сеть Интернет. Адресация в сети Интернет. Разделение IP-сети на подсети с помощью масок подсетей.

Умения и навыки: Наличие представлений о базовых принципах организации и функционирования компьютерных сетей. Умение использовать маску подсети.

Метапредметные результаты: *Работа с информацией* (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.) *Коммуникативные УУД* (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств). *Коммуникативные УУД* (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств).

Задание 14

Элементы содержания: Системы счисления. Развёрнутая запись целых и дробных чисел в позиционной системе счисления. Свойства позиционной записи числа: количество цифр в записи, признак делимости числа на основание системы счисления. Алгоритм перевода целого числа из Р-ичной системы счисления в десятичную. Алгоритм перевода целого числа из десятичной системы счисления в Р-ичную. Перевод конечной десятичной дроби в Р-ичную. Двоичная, восьмеричная и шестнадцатеричная системы счисления, связь между ними. Арифметические операции в позиционных системах счисления.

Умения и навыки: Умение использовать при решении задач свойства позиционной записи чисел, алгоритмы построения записи числа в позиционной системе счисления с заданным основанием и построения числа по строке, содержащей запись этого числа в позиционной системе счисления с заданным основанием; умение выполнять арифметические операции в позиционных системах счисления.

Метапредметные результаты: *Базовые логические действия* (Устанавливать существенный признак или основания для сравнения, классификации и обобщения. Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности. Развивать креативное мышление при решении жизненных проблем).

Задание 15

Элементы содержания: Алгебра логики. Понятие высказывания. Высказывательные формы (предикаты). Кванторы существования и всеобщности. Логические операции. Таблицы истинности. Логические выражения. Логические тождества. Логические операции и операции над множествами. Законы алгебры логики. Эквивалентные преобразования логических выражений.

Умения и навыки: Умение строить логическое выражение в дизъюнктивной и конъюнктивной нормальных формах по заданной таблице истинности; исследовать область истинности высказывания, содержащего переменные; решать несложные логические уравнения.

Метапредметные результаты: *Базовые логические действия* (Устанавливать существенный признак или основания для сравнения, классификации и обобщения. Выявлять закономерности и противоречия в рассматриваемых явлениях. Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. Вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности. Развивать креативное мышление при решении жизненных проблем). *Базовые исследовательские действия* (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов

действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов

Задание 16

Элементы содержания: Рекурсия. Рекурсивные процедуры и функции.

Умения и навыки: Вычисление рекуррентных выражений.

Метапредметные результаты: *Базовые логические действия* (Самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне; определять цели деятельности, задавать параметры и критерии их достижения. *Базовые исследовательские действия* (Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях.).

Задание 17

Элементы содержания: Массивы и последовательности чисел. Вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения, среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию). Линейный поиск заданного значения в массиве. Алгоритмы работы с элементами массива с однократным просмотром массива. Сортировка одномерного массива.

Умения и навыки: Умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива; умение использовать в программах данные различных типов с учётом ограничений на диапазон их возможных

значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; умение использовать средства отладки программ в среде программирования.

Метапредметные результаты: *Базовые исследовательские действия* (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов). *Регулятивные УУД 3.1 Самоорганизация* (Самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям. Самостоятельно составлять план решения проблемы с учётом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретённый опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний.) *3.2 Самоконтроль* (Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям. Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приёмы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению).

Задание 18.

Разделы/темы школьной программы: Анализ данных с помощью электронных таблиц. Вычисление суммы, сред-

него арифметического, наибольшего (наименьшего) значения диапазона. Решение задач оптимизации с помощью электронных таблиц.

Умения и навыки: Умение использовать электронные таблицы для анализа, представления и обработки данных (включая выбор оптимального решения, подбор линии тренда, решение задач прогнозирования); умение использовать табличные (реляционные) базы данных и справочные системы.

Метапредметные результаты: *Работа с информацией* (Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности).

Задание 20

Элементы содержания: Дискретные игры двух игроков с полной информацией. Построение дерева перебора вариантов, описание стратегии игры в табличной форме. Выигрышные и проигрышные позиции. Выигрышные стратегии.

Умения и навыки: Умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде. Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию.

Метапредметные результаты: *Работа с информацией* (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.). *Коммуникативные УУД* (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств).

Задание 21

Разделы/темы школьной программы: Дискретные игры двух игроков с полной информацией. Построение дерева перебора вариантов, описание стратегии игры в табличной форме. Выигрышные и проигрышные позиции. Выигрышные стратегии.

Умения и навыки: Умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать

адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде. Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию.

Метапредметные результаты: *Работа с информацией* (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.) *Коммуникативные УУД* (Развёрнуто и логично излагать свою точку зрения с использованием языковых средств).

Задание 22

Элементы содержания: Параллельные вычисления

Умения и навыки: Понимание основных принципов устройства и функционирования современных стационарных и мобильных компьютеров; тенденций развития компьютерных технологий; владение навыками работы с операционными системами и основными видами программного обеспечения для решения учебных задач по выбранной специализации. Построение математических моделей для решения практических задач. Архитектура современных компьютеров. Многопроцессорные системы.

Метапредметные результаты: *Работа с информацией* (Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления. Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.).

Задание 23

Элементы содержания: Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат.

Умения и навыки: Владение универсальным языком программирования высокого уровня (Паскаль, Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции; умение осуществлять анализ предложенной программы: определять результаты работы программы при

заданных исходных данных; определять, при каких исходных данных возможно получение указанных результатов; выявлять данные, которые могут привести к ошибке в работе программы; формулировать предложения по улучшению программного кода.

Метапредметные результаты: Базовые исследовательские действия (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов). *Регулятивные УУД 3.1 Самоорганизация* (Самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям. Самостоятельно составлять план решения проблемы с учётом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретённый опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний.) *3.2 Самоконтроль* (Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям. Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приёмы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению).

○ *Анализ типичных дефицитов в предметной подготовке: неосвоенные или слабо освоенные темы программы, несформированные или недостаточно сформированные умения, навыки, виды познавательной деятельности,*

проявляющиеся в типичных ошибках при выполнении соответствующих заданий

Таблица 2-19

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП
1.	Задание 24 <u>Элементы содержания:</u> Обработка символьных данных. Встроенные функции языка программирования для обработки символьных строк. Алгоритмы обработки символьных строк: подсчёт количества появлений символа в строке, разбиение строки на слова по пробельным символам, поиск подстроки внутри данной строки, замена найденной подстроки на другую строку. Генерация всех слов в некотором алфавите, удовлетворяющих заданным ограничениям. Преобразование числа в символьную строку и обратно. <u>Умения и навыки:</u> Владение универсальным языком программирования высокого уровня (Паскаль, Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции; умение осуществлять анализ предложенной программы: определять результаты работы программы при заданных исходных данных; определять, при каких исходных данных возможно получение указанных результатов; выявлять данные, которые могут привести к ошибке в работе программы; формулировать предложения по улучшению программного кода. <u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной	Базовый уровень: 11 кл. п. 113.7.3. Углубленный уровень: 10 кл., п. 114.6.3; 11 кл., п. 114.7.2

	деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов). <i>Регулятивные УУД 3.1 Самоорганизация</i> (Самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям. Самостоятельно составлять план решения проблемы с учётом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретённый опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний.) <i>3.2 Самоконтроль</i> (Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям. Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приёмы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению).	
2.	Задание 26 <u>Элементы содержания:</u> Вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения, среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию). Линейный поиск заданного значения в массиве. Алгоритмы работы с элементами массива с однократным просмотром массива. Сортировка одномерного массива. <u>Умения и навыки:</u> Умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива; умение использовать в программах данные различных типов с учётом ограничений на диапазон их возможных значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; умение использовать средства отладки программ в среде программирования.	Базовый уровень: 11 кл., п. 113.7.3. Углубленный уровень: 10 кл., п. 114.6.3; 11 кл., п. 114.7.2

<u>Метапредметные результаты:</u> <i>Базовые исследовательские действия</i> (Владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Овладение видами деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов. Формирование научного типа мышления, владение научной терминологией, ключевыми понятиями и методами. Выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения. Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду. Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов). <i>Регулятивные УУД 3.1 Самоорганизация</i> (Самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям. Самостоятельно составлять план решения проблемы с учётом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретённый опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний.) <i>3.2 Самоконтроль</i> (Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям. Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приёмы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению).	
--	--

- *Реалистичные «зоны роста» в части предметной подготовки для получения 100 т.б.*

На основе анализа данных Таблицы 2-13 для группы «от 81 до 100 т.б.», выделены задания, которые представляют собой наиболее реалистичные «зоны роста» для достижения 100 тестовых баллов: это задания с недостаточно высоким или низким показателем выполнения.

1. Задание № 24 (Высокий уровень)

Тема: Умение создавать собственные программы (10–20 строк) для обработки символьной информации.

Показатель	Значение
% выполнения в группе «81–100»	15,9%
Средний % выполнения	2,91%

Почему это «зона роста»: задание имеет крайне низкий процент выполнения даже среди высокобалльников (только 15,9% справляются). Основные ошибки связаны с неумением формализовать правила построения корректного выражения, отсутствием навыка использования регулярных выражений или ручного парсинга, а также недостаточным тестированием на граничных случаях. При целенаправленной отработке навыков обработки символьных строк и использования регулярных выражений задание становится решаемым.

Что отрабатывать:

- формализация правил построения корректного выражения на бумаге перед написанием кода;
- использование регулярных выражений (модуль `re` в Python): `finditer`, `search`, `match`, использование просмотра вперёд (`?=...`) для поиска перекрывающихся подстрок;
- ручной парсинг строк: посимвольный анализ, проверка структуры слева направо;
- тестирование на малых примерах: создание тестовых строк и проверка корректности работы программы;
- учёт всех ограничений: отсутствие ведущих нулей, недопустимость двух операций подряд, недопустимость операции в начале или конце выражения.

2. Задание № 25 (Высокий уровень)

Тема: Умение создавать собственные программы (10–20 строк) для обработки целочисленной информации.

Показатель	Значение
% выполнения в группе «81–100»	60,67%
Средний % выполнения	12,18%

Почему это «зона роста»: задание выполняется 60% высокобалльников, что оставляет резерв для улучшения. Основные ошибки связаны с непониманием условия (что множителей должно быть ровно два), неэффективными алгоритмами поиска делителей, неполной проверкой условий. При отработке эффективных алгоритмов разложения на простые множители и проверки делимости задание может быть выполнено практически всеми высокобалльниками.

Что отрабатывать:

разложение числа на простые множители: рекурсивная функция `def f(x): for i in range(2, int(x**0.5)+1): if x%i==0: return [i]+f(x//i) return [x];`

проверка, что в разложении ровно два простых множителя;

проверка, что каждый множитель содержит заданную подстроку (например, «43») ровно один раз;

оптимизация перебора: использование `range(2, int(x**0.5)+1)` для поиска делителей.

Тестирование на малых примерах перед запуском на больших числах.

3. Задание № 26 (Высокий уровень)

Тема: Умение обрабатывать целочисленную информацию с использованием сортировки.

Показатель	Значение
% выполнения в группе «81–100»	24,06%
Средний % выполнения	4,08%

Почему это «зона роста»: задание выполняется только 24% высокобалльников. Основные ошибки связаны с непониманием логики резервирования памяти, неправильным сравнением времени, неверным суммированием данных по клиентам, отсутствием проверки граничных условий. При систематической отработке алгоритмов обработки данных из файлов с использованием сортировки и словарей задание становится решаемым.

Что отрабатывать:

чтение данных из файла: `with open(filename, 'r') as file: n, k = map(int, file.readline().split());`

использование словарей для суммирования данных по ключам (идентификаторам клиентов);

логика резервирования: при `mem + s > k` сохранять `mem` (если время $\geq$ заданного порога) и присваивать `mem = s`;

сравнение времени в формате ЧЧ:ММ:СС (строковое сравнение работает корректно, если формат одинаковый);

сортировка списка по убыванию для нахождения двух наибольших значений;

проверка наличия хотя бы двух элементов в списке перед суммированием;

тестирование на примерах из условия.

4. Задание № 27 (Высокий уровень)

Тема: Умение выполнять последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и построение модели, преобразование данных, визуализация данных, интерпретация результатов.

Показатель	Значение
% выполнения в группе «81–100»	53,77%
Средний % выполнения	10,33%

Почему это «зона роста»: задание выполняется чуть более половины высокобалльников. Основные ошибки связаны с неправильной кластеризацией, неверным нахождением центров кластеров, невнимательным чтением условия (путаница между файлами А и В), забывают умножать результат на 10000 и находить целую часть. Задание требует системного подхода и внимательности к деталям, но при регулярной отработке прототипов становится решаемым.

Что отрабатывать:

кластеризация точек: использование алгоритма на основе расстояния между соседями (dist из модуля math);
 нахождение центра кластера: точка, сумма расстояний от которой до всех остальных точек кластера минимальна;
 работа с данными: чтение координат и классификационных признаков (спектральный класс, класс светимости);
 определение количества точек в кластерах;
 поиск ближайшей точки к центру кластера с заданными характеристиками (для файла А);
 вычисление расстояния между центрами кластеров с заданными характеристиками (для файла В);
 вывод результата с умножением на 10000 и взятием целой части.

Освоение указанных 4 тем (№ 24, 25, 26, 27) позволит учащимся из группы «от 81 до 100 т.б.» уверенно набрать 100 тестовых баллов. Все задания имеют стандартные алгоритмы решения, но требуют продвинутых навыков программирования, системного подхода и внимательности к деталям.

Рекомендуемая последовательность освоения:

1. Задания с чётким алгоритмом и возможностью тестирования на малых примерах (№ 25, 26).
2. Задания на обработку строк и регулярные выражения (№ 24).

3. Задания на кластеризацию и анализ данных (№ 27) – как наиболее объёмные и требующие системного подхода.

Особое внимание следует уделить:

тестированию на малых примерах перед запуском на больших данных;
пошаговой отладке с выводом промежуточных результатов;
внимательному чтению условия и выделению всех ограничений;
формализации задачи перед написанием кода.

3.2.4.2. Рекомендации для учителей по совершенствованию преподавания учебного предмета группе обучающихся с высоким уровнем подготовки

В отличие от других групп, где рекомендации фокусируются на освоении предметного содержания и ликвидации базовых дефицитов, работа с высокобалльниками должна быть направлена на развитие алгоритмического мышления высшего уровня, формирование навыков системного анализа и освоение эффективных приёмов программирования.

Рекомендация 1: Освоение «мета-алгоритмов» и комбинирование стандартных приёмов

Сильные обучающийся часто хорошо владеют отдельными алгоритмами (поиск, сортировка, перебор, динамическое программирование), но испытывают трудности в задачах, требующих комбинирования нескольких алгоритмических подходов в рамках одной программы (задания № 24, 26, 27).

Рекомендация (методика «Алгоритмический конструктор»):

1. Приём «Декомпозиция на независимые модули». Научите обучающихся разбивать сложную задачу на функционально независимые блоки. Например, для задания № 26:

Модуль 1: Чтение и парсинг данных из файла.

Модуль 2: Обработка логики резервирования и формирование списка копий.

Модуль 3: Агрегация данных по клиентам (словарь).

Модуль 4: Поиск клиента с максимальной суммой $\leq 150\,000$.

Модуль 5: Сортировка и поиск двух наибольших копий.

Каждый модуль должен быть протестирован отдельно на малых примерах перед объединением.

2. Приём «Комбинаторные связки». Предложите обучающимся задачи, где нужно применить два-три алгоритма последовательно. Например:

«Прочитайте данные, выполните кластеризацию, найдите центры кластеров, а затем среди точек каждого кластера найдите ближайшую к центру с заданными характеристиками» (прототип задания № 27).

Отрабатывайте навык переключения между структурами данных: список → словарь → множество → сортировка.

3. Приём «Смена парадигмы». Для одной и той же задачи предложите решить её тремя способами:

Чистое программирование (Python).

С использованием электронных таблиц (визуализация данных).

С использованием встроенных библиотек (itertools, collections, re, math).

Это формирует понимание, когда какой инструмент эффективнее, и развивает гибкость мышления.

Рекомендация 2: Формирование навыков анализа сложности и оптимизации кода

Даже сильные обучающийся часто пишут работающий, но неоптимальный код, который может не уложиться в ограничения по времени (особенно для заданий № 24–27 с большими объёмами данных – до 10 000 точек или 1 000 000 строк).

Рекомендация (методика «Профилирование и оптимизация»):

1. Приём «Оценка сложности до написания кода». Перед тем как начать писать программу, обучающийся должен оценить:

Ожидаемый объём входных данных (N, K).

Сложность алгоритма в терминах «О-большое».

Будет ли программа работать в пределах 1-2 секунд на типичном компьютере.

Пример для задания № 26: если $N = 1\,000\,000$, то решение должно работать за $O(N)$ или $O(N \log N)$, но не за $O(N^2)$. Перебор всех пар клиентов недопустим.

2. Приём «Использование встроенных функций вместо самописных». Категорически запрещайте обучающимся писать собственные реализации сортировки, поиска минимума/максимума, если есть встроенные аналоги. Примеры:

Вместо ручной сортировки – `data.sort(reverse=True)`.

Вместо поиска минимума в цикле – `min(data)`.

Вместо подсчёта частот вручную – `collections.Counter`.

Обсуждайте, что встроенные функции написаны на C и работают на порядок быстрее.

3. Приём «Оптимизация памяти». Учите работать с данными «на лету», не сохраняя всё в память, если это

возможно. Например, для задания № 17 (обработка последовательности) данные можно обрабатывать построчно, а не загружать весь файл в список.

```
python
# Вместо загрузки всех данных в список:
with open(filename) as f:
    n = int(f.readline())
    for _ in range(n):
        line = f.readline()
    # обработка строки сразу
```

4. Приём «Профилирование кода». Покажите обучающимся, как использовать модуль cProfile или time для замера времени выполнения отдельных участков программы. Это развивает «чувство» эффективности кода.

Рекомендация 3: Развитие навыков тестирования и верификации (стресс-тестирование)

Сильные обучающиеся часто проверяют решение только на примерах из условия, что недостаточно для выявления скрытых ошибок на граничных случаях (пустые файлы, минимальные/максимальные значения, нестандартный формат данных).

Рекомендация (методика «Индустриальное тестирование»):

1. Приём «Генератор случайных тестов». Научите обучающихся писать скрипты для генерации случайных тестовых данных. Например, для задания № 26:

Сгенерировать случайные N и K.

Сгенерировать случайные времена, идентификаторы клиентов и объёмы данных.

Запустить решение на сгенерированном тесте.

Написать «медленное, но гарантированно правильное» решение (брутфорс) и сравнить результаты.

Это позволяет отлавливать ошибки в логике, которые не видны на малых примерах.

2. Приём «Проверка на граничных условиях». Составьте список обязательных тестов для каждой задачи:

Минимальные значения (0, 1, пустой файл).

Максимальные значения (границы диапазонов).

Специфические случаи (например, для задания № 26: время ровно 12:00:00, клиент с суммой ровно 150 000, отсутствие подходящих клиентов).

Нестандартные случаи (например, раздел памяти полностью заполняется с первого запроса).

3. Приём «Инварианты и проверки». Научите добавлять в код assert для проверки инвариантов:

```
python
```

```
assert len(a) >= 2, "Недостаточно резервных копий"
```

```
assert sum_volumes <= k * n, "Некорректный расчёт"
```

Это помогает «поймать» ошибку на раннем этапе выполнения программы.

4. Приём «Визуализация промежуточных результатов». Для задач с геометрией или кластеризацией (№ 27) используйте модуль turtle или matplotlib, чтобы визуализировать кластеры, центры и расстояния. Это мгновенно показывает ошибки в логике разбиения точек.

Рекомендация 4: Работа с задачами на уровне «синтез» – проектная деятельность и олимпиадные задачи

Стандартных задач из ЕГЭ недостаточно для развития навыков синтеза и системного мышления, необходимых для выполнения заданий № 24–27 на 100 баллов. Обучающийся должен уметь «видеть» задачу в целом, а не как набор отдельных приёмов.

Рекомендация (методика «Синтез через проекты»):

1. Приём «Мини-проекты на основе реальных данных». Предложите обучающимся задачи, имитирующие реальные бизнес-ситуации:

«Обработайте лог-файл веб-сервера: определите самых активных пользователей, пиковые нагрузки, среднее время ответа».

«Проведите кластеризацию точек на карте города по типу объектов (кафе, магазины, школы) и найдите центры кластеров».

«Разработайте систему резервного копирования данных с ограниченным объёмом памяти».

Такие проекты требуют синтеза навыков: чтение данных, структурирование, анализ, визуализация.

2. Приём «Олимпиадные задачи сверх ЕГЭ». Включите в подготовку задачи с олимпиад по программированию (сайты: Codeforces, Informatics, «Сириус», «Тинькофф Поколение»), особенно на темы:

Динамическое программирование (задания № 23 – углублённо).

Обработка строк и регулярные выражения (задания № 24 – усложнённые варианты).

Анализ данных и кластеризация (задания № 27 – продвинутые).

После решения олимпиадной задачи обсуждайте, как её можно «сжать» до формата ЕГЭ и что для этого нужно изменить.

3. Приём «Рефакторинг и код-ревью». Организуйте практику «код-ревью» в группе сильных обучающихся.

Один обучающийся пишет решение, другой анализирует его на предмет:

Эффективности.

Читаемости (именование переменных, структура).

Надёжности (обработка ошибок, проверка границ).

Это развивает критическое мышление и способность видеть код с разных сторон.

Рекомендация 5: Формирование системного подхода к сложным многошаговым задачам

Задания № 24–27 требуют не только программирования, но и последовательного выполнения множества шагов (формализация, моделирование, кодирование, тестирование, интерпретация). Сильные обучающийся иногда пропускают этапы или выполняют их в неоптимальной последовательности.

Используйте методику «Чек-лист эксперта»:

1. Приём «Карта решения». Составьте для каждого задания высокого уровня подробную «карту решения» – последовательность шагов с указанием, что именно нужно делать на каждом этапе. Пример для задания № 27:

Этап	Действие	Критерий завершённости
1. Чтение данных	Прочитать файл, преобразовать координаты в float, сохранить в список с классификацией.	Все строки прочитаны, типы данных корректны.
2. Кластеризация	Разбить точки на кластеры по расстоянию $< d$.	Количество кластеров совпадает с ожидаемым (по условию).
3. Визуализация	Отобразить кластеры разными цветами (для самопроверки).	На экране видны чётко разделённые группы точек.
4. Поиск центров	Для каждого кластера найти центр (сумма расстояний минимальна).	Центры находятся внутри своих кластеров.
5. Обработка файла А	Найти кластер с наибольшим количеством точек, затем в нём – ближайшую к центру бело-голубую звезду.	Координаты звезды найдены.
6. Обработка файла В	Найти кластеры с наименьшим/наибольшим количеством жёлтых карликов, вычислить расстояние между центрами. Найти оранжевых карликов, вычислить максимальное расстояние внутри кластера.	Расстояния вычислены.
7. Вывод	Умножить результаты на 10000, взять целую часть, вывести в правильном формате.	Формат ответа соответствует требованию.

2. Обсуждайте с обучающимися, почему важен именно такой порядок действий.

3. Приём «Точки контроля». Для каждой задачи определите 2-3 промежуточных контрольных точки, где ученик может проверить правильность своих действий. Например, для задания № 26:

Контроль 1: словарь d содержит суммы всех клиентов.

Контроль 2: список a содержит все резервные копии, выполненные не ранее 12:00:00.

Контроль 3: mx и client выбраны корректно.

4. Приём «Анализ альтернативных решений». После того как обучающийся решил задачу, предложите ему найти другой способ решения (например, через электронные таблицы или с использованием другой структуры данных). Сравните оба подхода по сложности, времени написания и надёжности. Это формирует гибкость и готовность к нестандартным ситуациям.

5.

ВЫВОД

Работа с высокобалльниками должна выходить за рамки «натаскивания» на задания ЕГЭ и быть направлена на:

1. Развитие системного мышления – умение видеть задачу как единое целое и разбивать её на логические модули.
 2. Освоение продвинутых алгоритмических приёмов – комбинирование структур данных, оценка сложности, оптимизация.
 3. Формирование инженерной культуры программирования – тестирование, профилирование, рефакторинг, код-ревью.
 4. Развитие навыков синтеза – решение проектных задач, выходящих за рамки ЕГЭ, и олимпиадных задач.
- Эти рекомендации позволят обучающимся не просто получить 100 баллов на ЕГЭ, но и приобрести навыки, полезные для дальнейшего обучения в вузе и профессиональной деятельности в IT-сфере.

3.3. Рекомендации администрации ОО, ИПК / ИРО, иным организациям, реализующим программы профессионального развития учителей по совершенствованию преподавания учебного предмета

3.3.1. Рекомендуемые темы для обсуждения / обмена опытом на школьных методических объединениях учителей-предметников, в том числе по трансляции эффективных педагогических практик ОО со стабильно высокими результатами ЕГЭ

Роль математического моделирования в обучении информатике;

Практика обучения обработке числовых данных средствами электронных таблиц;

Метод динамического программирования и рекурсия в задачах ЕГЭ;

Формирование навыков самоорганизации и самоконтроля в обучении информатике и подготовке к ЕГЭ;
Однопараметрические и многопараметрические методы сортировок данных.

3.3.2. Рекомендуемые ИПК / ИРО, иным организациям, реализующим программы профессионального развития учителей, направления повышения квалификации (в форме курсов, отдельных мероприятий, консультаций и т.п.) с учетом выявленных дефицитов предметной подготовки

Для повышения качества подготовки к ЕГЭ по информатике наиболее актуальны курсы и консультации, фокусирующиеся на практической отработке сложных алгоритмических задач и развитии у обучающихся метапредметных навыков. Эффективной формой работы станут регулярные методические вебинары, семинары и обмен опытом между учителями.

Основываясь на выявленных в отчете дефицитах, можно предложить следующие направления для повышения квалификации и методической поддержки учителей информатики.

Курсы повышения квалификации

1. Практикум по решению проблемных заданий ЕГЭ – программы, направленные на разбор задач повышенного и высокого уровня сложности (в частности, игровых задач, обработки массивов и последовательностей) с отработкой алгоритмов решения и методик предотвращения типичных ошибок.
2. Развитие метапредметных компетенций – программы, направленные на формирование у обучающихся навыков самоорганизации, самоконтроля и анализа условия задач, что является ключевым фактором успеха.

Тематика консультаций и методических мероприятий

1. Разбор сложных заданий ЕГЭ-2026 – детальный анализ новых заданий и изменений в КИМ, а также стратегий подготовки к ним.
2. Методика обучения решению игровых задач – разбор построения дерева игры и поиска выигрышных стратегий.
3. Методика обучения алгоритмизации и программированию – отработка практических навыков работы с файлами, массивами, сортировкой и обработкой данных.
4. Трансляция эффективных педагогических практик – обмен опытом между учителями школ с высокими результатами.

Рекомендации по развитию профессиональных компетенций учителей

1. Развитие практических навыков программирования: Для эффективной подготовки, особенно к заданиям с

развернутым ответом (№24–27), учителю необходимо регулярно совершенствовать свои навыки в практическом программировании, включая работу с файлами, массивами и структурами данных.

2. Освоение методик формирования метапредметных умений: Учителям стоит уделить внимание тому, как научить обучающихся "читать" условия задач, вычленять ключевые данные, планировать решение и проверять результаты, то есть развивать метапредметные компетенции.

3. Использование современных педагогических технологий и мотивации: Важно внедрять интерактивные и проектные методики, чтобы повысить мотивацию обучающихся, а также использовать дифференцированный подход в обучении.

3.3.3. Рекомендации по другим направлениям (при наличии) нет

СОСТАВИТЕЛИ ОТЧЕТА по учебному предмету:

Специалисты, привлекаемые к анализу результатов ЕГЭ по учебному предмету

<i>Фамилия, имя, отчество</i>	<i>Место работы, должность, ученая степень, ученое звание, принадлежность специалиста (к региональным организациям развития образования, к региональным организациям повышения квалификации работников образования, к региональной ПК по учебному предмету, пр.)</i>
Афониная Марина Викторовна	ФГБОУ ВО «АлтГПУ», доцент кафедры теоретических основ информатики, кандидат педагогических наук, председатель предметной комиссии ОГЭ по информатики в Алтайском крае

Специалисты, привлекаемые к проведению методического анализа результатов ЕГЭ по учебному предмету и рекомендаций для региональной системы образования

<i>Фамилия, имя, отчество</i>	<i>Место работы, должность, ученая степень, ученое звание, принадлежность специалиста (к региональным организациям развития образования, к региональным организациям повышения квалификации работников образования, к региональной ПК по учебному предмету, пр.)</i>
Афониная Марина Викторовна	ФГБОУ ВО «АлтГПУ», доцент кафедры теоретических основ информатики, кандидат педагогических наук, председатель предметной комиссии ОГЭ по информатики в Алтайском крае
Чеверда Ирина Викторовна	заместитель директора КАУ ДПО «Алтайский институт развития образования имен Адриана Митрофановича Топорова»

Ответственный специалист в субъекте Российской Федерации по вопросам организации проведения анализа результатов ЕГЭ по учеб-

ным предметам

<i>Фамилия, имя, отчество</i>	<i>Место работы, должность, ученая степень, ученое звание</i>
Чибрякова Татьяна Евгеньевна	консультант отдела организации общего образования и оценочных процедур Министерства образования и науки Алтайского края