

**Министерство образования Иркутской области
Государственное автономное учреждение Иркутской области
«Центр оценки профессионального мастерства, квалификаций педагогов и мониторинга
качества образования»**

**Методический анализ результатов ЕГЭ по ИНФОРМАТИКЕ
в Иркутской области в 2025 году**

Иркутск, 2025

Методический анализ результатов ЕГЭ¹ по информатике

Раздел 1. АНАЛИЗ РЕЗУЛЬТАТОВ ВЫПОЛНЕНИЯ ЗАДАНИЙ КИМ

1.1 Анализ выполнения заданий КИМ

1.1.1 Статистический анализ выполнения заданий КИМ в 2025 году

Анализ выполнения КИМ в разделе 3.1 выполняется на основе результатов всего массива участников основного периода ЕГЭ по информатике в Иркутской области вне зависимости от выполненного участником экзамена варианта КИМ.

1.1.1.1 Основные статистические характеристики выполнения заданий КИМ в 2025 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.

¹ Для анализа использовался массив результатов основного дня основного периода ЕГЭ

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Иркутской области ² 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.
1.	Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)	Б	81,9	59,3	90,3	94,0	97,1
2.	Умение строить таблицы истинности и логические схемы	Б	64,3	24,5	72,8	93,5	99,1
3.	Умение поиска информации в реляционных базах данных	Б	69,9	41,5	77,9	88,0	92,4
4.	Умение кодировать и декодировать информацию	Б	80,9	55,2	89,3	96,1	99,5
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	Б	30,9	2,2	15,4	74,0	97,1
6.	Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов	Б	37,3	5,2	34,0	69,8	87,6
7.	Умение определять объём памяти, необходимый для хранения графической и звуковой информации	Б	51,8	13,2	53,6	86,9	96,2
8.	Знание основных понятий и методов, используемых при измерении количества информации	Б	36,0	2,1	26,0	79,3	95,2
9.	Умение обрабатывать числовую информацию в электронных таблицах	Б	18,8	1,0	9,1	38,0	80,5
10.	Информационный поиск средствами текстового процессора	Б	73,8	52,7	79,6	86,8	92,4

² Вычисляется по формуле $p = \frac{N}{n \cdot m} \cdot 100\%$, 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 Иркутской области ² 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	П	21,4	2,2	14,0	40,8	76,7
12.	Умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд	П	32,5	1,6	21,7	73,5	91,0
13.	Умение использовать маску подсети	П	30,5	1,9	18,8	67,9	93,3
14.	Знание позиционных систем счисления	П	34,1	2,2	22,8	76,2	95,7
15.	Знание основных понятий и законов математической логики	П	37,7	3,7	27,2	81,5	98,6
16.	Вычисление рекуррентных выражений	П	43,7	7,2	43,4	76,9	95,7
17.	Умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования	П	15,4	0,0	2,3	34,1	85,2
18.	Умение использовать электронные таблицы для обработки целочисленных данных	П	30,9	3,5	24,8	57,8	91,9
19.	Умение анализировать алгоритм логической игры	Б	59,4	24,4	61,1	91,5	99,5
20.	Умение найти выигрышную стратегию игры	П	39,8	2,2	32,6	84,3	98,1
21.	Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию	В	37,9	1,5	28,8	82,7	99,1
22.	Построение математических моделей для решения практических задач. Архитектура современных компьютеров. Многопроцессорные системы	П	25,3	4,1	20,1	43,3	81,9
23.	Умение анализировать ход исполнения алгоритма	П	36,5	2,5	24,6	83,0	96,7
24.	Умение создавать собственные программы (10–20 строк) для обработки символьной информации	В	3,5	0,0	0,2	2,7	34,3

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Иркутской области ² 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.
25.	Умение создавать собственные программы (10–20 строк) для обработки целочисленной информации	В	5,3	0,0	0,0	6,5	46,7
26.	Умение обрабатывать целочисленную информацию с использованием сортировки	В	3,0	0,0	0,0	2,0	34,0
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	В	7,0	0,0	0,2	9,0	60,0

Таблица 0-2

Номер задания / критерия оценивания в КИМ	Количество полученных первичных баллов	Процент участников экзамена в Иркутской области, 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., %
1.	Не приступал	1,4	0,5	0,4	0,0
	0	39,4	9,2	5,7	2,9
	1	59,3	90,3	94,0	97,1
2.	Не приступал	4,2	0,6	0,2	0,0
	0	71,3	26,6	6,4	0,9
	1	24,5	72,8	93,5	99,1

Номер задания / критерия оценивания в КИМ	Количество полученных первичных баллов	Процент участников экзамена в Иркутской области, 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., %
3.	Не приступал	4,4	0,7	0,7	0,0
	0	54,2	21,4	11,3	7,6
	1	41,5	77,9	88,0	92,4
4.	Не приступал	1,7	0,0	0,0	0,0
	0	43,1	10,8	3,9	0,5
	1	55,2	89,3	96,1	99,5
5.	Не приступал	39,6	34,7	6,0	0,5
	0	58,1	50,0	20,0	2,4
	1	2,2	15,4	74,0	97,1
6.	Не приступал	21,7	9,7	1,4	0,0
	0	73,0	56,3	28,8	12,4
	1	5,2	34,0	69,8	87,6
7.	Не приступал	14,5	5,2	0,5	0,0
	0	72,3	41,2	12,5	3,8
	1	13,2	53,6	86,9	96,2
8.	Не приступал	26,6	17,8	1,4	0,0
	0	71,3	56,2	19,3	4,8
	1	2,1	26,0	79,3	95,2
9.	Не приступал	45,3	44,9	12,9	0,5
	0	53,7	46,0	49,1	19,1
	1	1,0	9,1	38,0	80,5
10.	Не приступал	2,1	0,1	0,0	0,0
	0	45,2	20,3	13,3	7,6
	1	52,7	79,6	86,8	92,4

Номер задания / критерия оценивания в КИМ	Количество полученных первичных баллов	Процент участников экзамена в Иркутской области, 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., %
11.	Не приступал	26,8	14,1	2,1	0,0
	0	70,9	71,9	57,1	23,3
	1	2,2	14,0	40,8	76,7
12.	Не приступал	49,2	26,5	2,7	0,0
	0	49,2	51,8	23,9	9,1
	1	1,6	21,7	73,5	91,0
13.	Не приступал	43,7	31,7	5,7	0,5
	0	54,4	49,4	26,5	6,2
	1	1,9	18,8	67,8	93,3
14.	Не приступал	55,8	39,0	4,2	0,0
	0	42,0	38,2	19,6	4,3
	1	2,2	22,8	76,2	95,7
15.	Не приступал	50,2	35,9	6,9	0,0
	0	46,1	36,9	11,7	1,4
	1	3,7	27,2	81,5	98,6
16.	Не приступал	49,2	30,6	9,0	1,0
	0	43,6	26,0	14,1	3,3
	1	7,2	43,4	76,9	95,7
17.	Не приступал	70,1	71,0	29,7	0,0
	0	29,9	26,7	36,2	14,8
	1	0,0	2,3	34,1	85,2
18.	Не приступал	47,7	23,1	3,7	0,0
	0	48,8	52,1	38,5	8,1
	1	3,5	24,8	57,8	91,9

Номер задания / критерия оценивания в КИМ	Количество полученных первичных баллов	Процент участников экзамена в Иркутской области, 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., %
19.	Не приступал	27,7	9,7	0,7	0,0
	0	48,0	29,2	7,8	0,5
	1	24,4	61,1	91,5	99,5
20.	Не приступал	39,3	15,3	1,2	0,0
	0	58,5	52,2	14,5	1,9
	1	2,2	32,6	84,3	98,1
21.	Не приступал	43,1	21,4	1,8	0,0
	0	55,4	49,9	15,6	1,0
	1	1,5	28,8	82,7	99,1
22.	Не приступал	52,7	35,9	12,2	1,9
	0	43,2	44,0	44,5	16,2
	1	4,1	20,1	43,3	81,9
23.	Не приступал	53,8	33,9	5,7	0,0
	0	43,7	41,6	11,3	3,3
	1	2,5	24,6	83,0	96,7
24.	Не приступал	72,6	78,3	64,0	22,4
	0	27,5	21,5	33,4	43,3
	1	0,0	0,2	2,7	34,3
25.	Не приступал	90,3	92,8	73,1	18,1
	0	9,7	7,2	20,3	35,2
	1	0,0	0,0	6,5	46,7
26.	Не приступал	82,6	86,4	75,1	31,0
	0	17,4	13,5	22,3	24,3
	1	0,0	0,1	1,9	21,0

Номер задания / критерия оценивания в КИМ	Количество полученных первичных баллов	Процент участников экзамена в Иркутской области, 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., %
	2	0,0	0,0	0,7	23,8
27.	Не приступал	84,7	88,3	70,9	15,2
	0	15,2	11,3	15,7	8,1
	1	0,1	0,2	9,4	33,3
	2	0,0	0,2	4,1	43,3

1.1.1.2 Выявление сложных для участников ЕГЭ заданий

- Задания базового уровня (с процентом выполнения ниже 50):
 - Задание 5 (30,9%)
 - Задание 6 (37,3%)
 - Задание 8 (36,0%)
 - Задание 9 (18,8%)
- Задания повышенного и высокого уровня (с процентом выполнения ниже 15)
 - Задание 24 (3,5%)
 - Задание 25 (5,3%)
 - Задание 26 (2,1%)
 - Задание 27 (7,0%)
- Прочие задания
 - Задание 11 (21,4%; каждый год вызывает затруднения)
 - Задание 12 (32,5%; резкое снижение процента выполнения)
 - Задание 17 (15,4%)

1.1.1.3 Прочие результаты статистического анализа

Рассмотрим дополнительно успешность выполнения заданий в сравнении с 2023 и 2024 годами, а также по вариантам, отправленным в муниципальные образования. Для краткости все выделенные группы обозначим номерами: группа 1 – экзаменуемые, не преодолевшие минимальный балл; группа 2 – набравшие от минимального до 60 тестовых баллов; группа 3 – набравшие от 61 до 80 тестовых баллов; группа 4 – набравшие от 81 до 100 баллов.

На базовом уровне будем оценивать успешность выполнения заданий, если средний процент его выполнения больше 70, на повышенном – выше 40, на высоком – выше 30.

На базовом уровне в среднем по области наиболее успешно были выполнены задания 1 (81,9%), 4 (80,9%) из содержательного раздела «Теоретические основы информатики», а также задание 10 (73,8%) из раздела «Информационные технологии».

Успешность выполнения задания 1 осталась практически на том же уровне, как и в 2024 году, но по сравнению с 2023 годом этот показатель уменьшился на 7%. Традиционно с этим заданием справляются практически все выпускники, попавшие в 4 группу (более 97%), в остальных группах процент выполнения колеблется незначительно (около 90% для групп 2 и 3 и около 60% для группы 1).

Средний процент выполнения задания 2 («Теоретические основы информатики») снизился по сравнению с 2024 годом сразу на 10%. Это снижение произошло за счет учащихся, попавших в группы 1, 2 и 3. Выпускники с высоким уровнем подготовки справляются с ним традиционно хорошо (99,1%).

Значительно ухудшились показатели успешности выполнения задания 5 (раздел «Алгоритмы и программирование»). Так, в 2024 году в среднем по области с ним успешно справились 44% выпускников, а в 2025 году этот процент упал до 30,9%. Снижение произошло более, чем в два раза для групп 1 и 2. В группе 4, наоборот, этот процент увеличился с 95,7% до 97,1%. Такие показатели можно объяснить, в частности, тем, что по сравнению с 2024 годом в формулировках задания речь шла не только о привычной для этого задания двоичной системе счисления, но и о других системах. Поэтому учащиеся со слабой подготовкой это задание выполнить правильно не смогли. Стоит отметить, что те выпускники, которые решали вариант 362, продемонстрировали значительный рост успешности выполнения этого задания (они в 44,1% справились с ним). В других вариантах этот процент либо близок к среднему

по области, либо снижен. Вероятно, формулировка этого задания в варианте 362 оказалась проще и понятнее (или привычней), чем в других вариантах.

Задание 6 («Алгоритмы и программирование») в 2025 году было выполнено недостаточно успешно, но по сравнению с 2024 годом процент выполнения в среднем по области увеличился (с 34% до 37,3%). Таким образом, исключение из регионального перечня программного обеспечения школьного алгоритмического языка и среды Кумир не повлияло на результаты выполнения этого задания. Те учащиеся, которые владеют аналитическим или программным способом решения этого задания (а это выпускники, попавшие во все группы, кроме 1), продемонстрировали рост успешности его выполнения в сравнении с 2024 годом. Выполнение этого задания теми, кто решал варианты 325, 327, 330 и 363, просело более, чем на 10% по сравнению со средним процентом по области. Зато те, кто выполнял вариант 361, справились с ним лучше (рост более, чем на 10%).

К сожалению, резкое снижение процента успешности выполнения в 2025 году произошло с заданием 9 (раздел «Информационные технологии»), если в 2024 году с ним успешно справились 33,1% выпускников, то в 2025 году этот показатель составил всего 18,8%. Причем снижение отмечается во всех группах подготовки, даже в группе 4 этот процент упал с 97% в 2024 году до 80,5% в 2025. Вероятно, это задание было изменено в части постановки вопроса и те, кто привык решать эту задачу в старой формулировке, по невнимательности выполнили его неверно (что опять возвращает нас к вопросу о «натаскивании» и решении типовых задач готовыми шаблонами). Лучше других с заданием 9 справились те выпускники, которые выполняли варианты 361 и 362. Ниже нами будет рассмотрен пример формулировки этого задания с измененной постановкой задачи.

Задание 19 базового уровня сложности (раздел «Теоретические основы информатики»), являющееся частью цепного задания 19-21, в 2025 году продемонстрировало значительное снижение успешности выполнения в среднем по области (в 2024 году с ним справились 67,9%, а в 2025 – только 59,4%). Снижение произошло во всех четырех группах подготовки. Вероятно, это связано с другой системой команд, описанной в задании. Например, камни следовало не добавлять, а убирать из кучи, или при делении округлять не в меньшую, а в большую сторону. Такая динамика прослеживается и в задании 20 (повышенный уровень), и в задании 21 (высокий уровень). Все они показывают снижение успешности выполнения этого цепного задания. Такая тенденция наблюдается во всех вариантах, отправленных в регион.

Проанализируем задания повышенного уровня сложности. Из заданий, попадающих в эту категорию, наиболее успешно были выполнено только задание 16 (43,7%) содержательной линии «Алгоритмы и программирование». Остальные задания повышенного уровня были выполнены менее, чем на 40%. То есть, по сравнению с 2024 годом, эти задания были выполнены намного хуже: если в прошлом году в этот список попали три задания, то в текущем – всего одно.

Успешность выполнения задания 11 («Теоретические основы информатики») резко снизилась в 2024 году (с 49% в 2023 году до 12,8% в 2024). В 2025 году этот показатель несколько увеличился (до 21,4%), но все равно недостаточно высок. С этим заданием недостаточно успешно справляются выпускники всех групп подготовки (даже в группе 4 этот показатель составил 76,7%). Причины этого, скорее всего, те же, что и для других рассмотренных выше заданий: недостаточная предметная подготовка, шаблонность решения типовых задач, недостаточно сформированная читательская грамотность. Выпускники, выполнявшие вариант 365, показали самое серьезное падение успешности выполнения этого задания (при среднем показателе 21,4% они выполнили задание 11 с успешностью 10,7%). Зато у тех, кому достался вариант 366, это задание больших затруднений не вызвало (41,2%).

Очень серьезную тревогу вызывает ухудшение ситуации с выполнением задания 12 (раздел «Алгоритмы и программирование»): в 2025 году процент выполнения этого задания в среднем по области упал на 25% (с 57,5% в 2024 году до 32,5% в 2025 году). Причем снижение наблюдается во всех уровнях подготовки: в группе 1 – с 8% до 1,6%, в группе 2 – с 60,5% до 21,7%, в группе 3 – с 93% до 73,5%, даже выпускники, попавшие в группу 4, продемонстрировали снижение процента выполнения задания с 98,8% до 91%. Наихудшие результаты выполнения этого задания у тех выпускников, которые выполняли варианты 364 и 365.

Успешность выполнения задания 16 («Алгоритмы и программирование») также снизилась (43,7%) и стала самой низкой за три последних года. Этот показатель снизился во всех группах подготовки, кроме первой. В этой группе процент стал выше (с 2,5% в 2024 году до 7,2% в 2025 году). Как нам кажется, выпускников этой группы научили решать это задание аналитически, с помощью математических формул. Те, кто попал в остальные группы, скорее всего попытались решить это задание программированием, но из-за больших чисел не смогли реализовать рекурсивный алгоритм. Снижение успешности выполнения этого задания более, чем на 20% отмечается среди тех, кто

писал варианты 361, 362, 363, 365 и 366. А в вариантах 325, 326 и 327 это задание оказалось проще (успешность в среднем на 20% выше, чем по области).

Задание 18 (раздел «Информационные технологии») также продемонстрировало снижение процента успешности по сравнению с 2024 годом (с 42,3% до 30,9%), причем опять это снижение произошло во всех четырех группах. В 2024 году был отмечен достаточно неплохой рост успешности выполнения этого задания, но в 2025 году, видимо, опять появились немного измененные формулировки. Исключением является задание из 362 варианта – в нем процент выполнения задания 18 составил 44,7%, что на 14% выше, чем в среднем по области.

Успешность выполнения задания 23 (раздел «Алгоритмы и программирование») также стала ниже (36,5%, самый низкий процент за три последних года). Это снижение наблюдается во всех четырех группах. Вероятно, вновь была немного изменена формулировка задачи или система команд исполнителя.

Перейдем к заданиям высокой степени сложности. Среди них самым успешным является задание 21 (раздел «Теоретические основы информатики»), являющийся частью цепного задания 19-21. Успешность выполнения этого задания в 2025 году составила 37,9%, этот показатель является самым низким за три последних года. Эта величина не снизилась только для выпускников, попавших в четвертую группу подготовки. Вероятные причины этого рассмотрены выше при описании задания 19.

Остальные задания этой степени сложности были выполнены с очень низким процентом успешности – в среднем по области этот процент не превышает 6%.

Задание 24 (раздел «Алгоритмы и программирование») было выполнено с показателем успешности 3,5%. Выпускники, попавшие в группы 1 и 2, с этим заданием практически не справились (0,0% и 0,2% соответственно); участники группы 3 справились с ним на 2,7%, это немного повыше, чем в 2024 году, но значительно ниже показателя 2023 года. Также увеличился процент выполнения этого задания учащимися из четвертой группы (с 15,98% до 34,4%). Это связано с тем, что в 2024 году появились новые формулировки этого задания, направленные на анализ математических выражений, поэтому в 2024 году процент выполнения резко упал (с 6,3% до 1,8%). К 2025 году выпускники с хорошей предметной подготовкой решали подобные задачи, поэтому процент стал немного выше. Экзаменуемые, которые выполняли варианты 328 и 330, справились с этим заданием примерно на 3% лучше, чем в среднем по области, но задача из варианта 366 оказалась очень сложной: с ней справились только 1,6% выпускников.

Задание 25 (раздел «Алгоритмы и программирование») также было выполнено с низким процентом успешности, в среднем по области эта величина составила всего 5,3%. В 2024 году в это задание вновь вернулись задачи на делители, которые значительно сложнее, чем задачи на маски из 2023 года. Поэтому в прошлом году резко упали показатели успешности выполнения этого задания во всех четырех группах, а в этом году ситуация стала еще хуже: учащиеся из первой и второй групп вообще не справились с этим заданием (0,0% в обеих группах), учащиеся третьей группы дали величину 6,5% против 33,1% в 2024 году. Даже в группе 4 этот показатель значительно просел – с 77,4% в 2024 году до 46,7% в 2025 году. Вполне вероятно, что учащиеся даже с высоким уровнем подготовки не всегда могут написать эффективную программу, которая дает ответ за разумное время при работе с большими числами. К тому же, вполне вероятно и некоторое изменение формулировки в сторону усложнения. Ниже мы это покажем. Что касается вариантов, то самой сложной (1,7%) оказалась задача из варианта 325, напротив, задача из варианта 362, оказалась легче (10,1%), чем в других вариантах, отправленных в регион.

Задание 26 из раздела «Информационные технологии» тоже было выполнено хуже, чем в прошлом году (в 2024 году – 4,1%, в 2025 году – 3,2%). Этот показатель тоже стал самым низким за три последних года. Выпускники, попавшие в группы 1 и 2, традиционно с ним не справились практически полностью, показатель успешности в третьей группе составил 1,7% против 3,8% в 2024 году. Учащиеся четвертой группы выполнили его тоже хуже, чем в прошлом году (34,3% против 44,2%). Единственным исключением являются те, кто выполнял задания варианта 364 – они справились с этим заданием с показателем 6,9% против среднего по области.

Наконец, рассмотрим успешность выполнения задания 27 из содержательной линии «Алгоритмы и программирование». В 2025 году это задание обновилось полностью: изменились проверяемые предметные требования к результатам освоения ООП и проверяемые элементы содержания. Задание, на наш взгляд, стало легче, понятнее и даже интереснее. Это задание можно отнести к элементам машинного обучения, которое сейчас очень актуально. Поэтому и результаты выполнения этого задания, несмотря на его новизну, стали лучше: в 2024 году в среднем по области процент выполнения составил всего 2,4%, а в 2025 году он составил 4,5%. Конечно, учащиеся с низким уровнем подготовки (группы 1 и 2) с ним не справились, зато выпускники, попавшие в третью и четвертую группы, продемонстрировали значительный рост успешности его выполнения: для учащихся группы 3 – от 2,9% до 9,0%, а для группы 4 – от 23,5% до 60%.

Далее дадим сводный краткий анализ успешности выполнения задания для всех содержательных линий.

Содержательная линия «Цифровая грамотность» в КИМ представлена заданиями 13 и 22. Оба этих задания являются относительно новыми, показатели выполнения этих заданий в общем повышаются. Относительно невысокие показатели в заданиях этого раздела (особенно для групп 1 – 3) могут быть связаны с нехваткой базовых знаний в соответствующем разделе, нехваткой навыков подготовки и выполнения конкретных заданий, нехваткой навыков анализа данных и формулировки обоснованных выводов.

С заданиями из раздела «Теоретические основы информатики» (задания 1, 2, 4, 7, 8, 11, 14, 15, 19-21) по большей части справляются достаточно хорошо, в частности потому, что они в основном относятся к базовому уровню сложности. Но все равно по отдельным темам из этого раздела экзаменуемые подготовлены достаточно слабо, на что необходимо обратить внимание, особенно экзаменуемым из групп 1 и 2.

Показатели выполнения заданий раздела «Информационные технологии» (это задания 3, 9, 10, 18, 26) в целом держатся на одном уровне. Важно дальше повышать уровень освоения знаний и навыков, представленных в этом разделе, так как они применяются в любой сфере деятельности.

Последний содержательный раздел «Алгоритмы и программирование» содержит задания всех уровней сложности: задания 5 и 6 относятся к базовому уровню, задания 12, 16, 17, 23 – к повышенному, задания 24, 25 и 27 – к высокому. Отчетливо видно, что знания и навыки в этом разделе до сих пор недостаточно сформированы, хотя уже гораздо лучше, чем несколько лет назад. Повышение успешности выполнения части заданий из этого раздела можно связать с переходом на компьютерный вариант ЕГЭ по информатике. Экзаменуемые знают, какие языки и среды программирования они смогут использовать на экзамене, поэтому заранее начинают готовиться, получая сначала базовые знания в этом разделе, потом осваивая базовые навыки программирования, изучая базовые алгоритмы, а потом уже применяя их для решения конкретных заданий. Из статистики выполнения заданий отчетливо видно, что с заданиями базового и повышенного уровней сложности могут справляться экзаменуемые всех групп подготовки. Наиболее западает в группах 1 и 2 задание 17, но здесь мы ни в коем случае не советуем отказываться от включения подобных задач в программу обучения этих групп, так как там проверяется как раз умение применять самые простые, самые базовые алгоритмы программирования. К тому же, задание 17 можно решать и средствами электронных таблиц, а их использование на базовом уровне дается ребятам достаточно хорошо.

Из приведенных статистических данных можно сделать вывод, что для группы 1 совсем неосвоенными или плохо освоенными являются задания, связанные с алгоритмизацией и программированием, использованием сложных формул в электронных таблицах, определением количества информации, работой с различными системами счисления, заданиями из раздела «Цифровая грамотность», соответственно на это необходимо обратить внимание при разработке рекомендаций для подготовки экзаменуемых этой группы.

1.1.2 Содержательный анализ выполнения заданий КИМ

Рассмотрим задания, вызвавшие сложности при выполнении у обучающихся различного уровня подготовки, а также возможные ошибки и причины затруднений при их выполнении. Начнем с заданий базового уровня. По нашему глубокому убеждению, задания этого уровня сложности должны уметь решать обучающиеся всех уровней подготовки, конечно, при этом возможны и ошибки.

Задание 5 (содержательный раздел «Алгоритмы и программирование», средний процент выполнения 30,9%).

Пример формулировки³:

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

1. Строится троичная запись числа N .

2. Далее эта запись обрабатывается по следующему правилу:

а) если число N делится на 3, то к этой записи дописываются две последние троичные цифры;

б) если число N на 3 не делится, то остаток от деления умножается на 5, переводится в троичную запись и дописывается в конец числа. Полученная таким образом запись является троичной записью искомого числа R .

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

Например, для исходного числа $11_{10} = 102_3$ результатом является число $102101_3 = 307_{10}$, а для исходного числа $6_{10} = 20_3$ это число $2020_3 = 60_{10}$.

³ Здесь и далее используются примеры формулировок, взятых из открытого варианта КИМ, направленного в регион.

Укажите **минимальное** число N , после обработки которого с помощью этого алгоритма получается число R , не меньшее 242.

Сложности и ошибки, возникшие при решении задачи:

1. Ошибки при выполнении этого задания традиционно связаны с тем, что ученики недостаточно вдумчиво читают условие задачи и в качестве ответа записывают не то, что сказано в условии. Например, для приведенного задания в качестве ответа были записаны:

а. Число 27 – это значение исходного числа при $R=243$, то есть первого числа, соответствующего условию $R \geq 242$.

б. Число 102101 – это значение числа R в троичной системе счисления.

с. Число 17 – значение N , если вместо троичной системы счисления использовалась двоичная.

д. Числа, связанные с неверным выделением двух последних цифр из троичного числа.

е. И другие, индивидуальные ошибки.

2. Важно, что 27,1% выпускников вообще не приступили к выполнению этого задания. Это достаточно высокий процент для задания базового уровня.

Наибольшие сложности это задание вызывает в группах 1 и 2, хотя и в группе 4 процент выполнения этого задания не превышает 90%. Поэтому необходимо обратить внимание на базовую подготовку обучающихся по разделу «Алгоритмы и программирование» в соответствии с обновленным ФГОС, ни в коем случае не пропускать темы этого раздела, максимально следовать федеральной рабочей программе сначала для 7–9-х классов, а затем и для 10–11-х.

Посмотрим, какие темы, предусмотренные в федеральной рабочей программе ООО для 7–9-х классов базового уровня и в федеральной рабочей программе СОО для 10–11-х классов базового уровня, необходимо пройти для того, чтобы обучающиеся готовы были выполнить это задание:

8-й класс: «Системы счисления», 6 часов;

8-й класс: «Исполнители и алгоритмы. Алгоритмические конструкции», 10 часов;

8-й класс: «Анализ алгоритмов», 2 часа;

8-й класс: «Язык программирования», 9 часов, дополнительно;

9-й класс: «Разработка алгоритмов и программ», 6 часов, дополнительно;

10-й класс: «Представление информации в компьютере», 8 часов;

11-й класс: «Алгоритмы и элементы программирования», 11 часов, дополнительно.

Решать это задание можно различными способами: с помощью электронных таблиц, составлением программы. Способ решения зависит как от уровня подготовки, так и от того, сколько заданий экзаменуемый планирует решить и сколько баллов набрать. Для решения задачи 5 для начала необходимо научить обучающихся выполнять и анализировать линейный алгоритм, о чем и говорится в спецификации к этому заданию. Для этой цели хорошо подходят различные задачи на действия с числами или строками. Также важно при изучении темы «Системы счисления» показать, что можно выполнять перевод между некоторыми системами счисления для обучающихся с разным уровнем подготовки с использованием различных инструментов: калькулятор, электронные таблицы, языки программирования. Для решения задачи 5 полезно обладать навыком увеличения числа в позиционной системе счисления на 1, сравнения чисел в различных позиционных системах счисления.

На черновике, вручную, подобное задание решать не рекомендуется, так как могут возникнуть трудности с безошибочным применением описанного в условии задачи алгоритма.

Задание конкретно в такой формулировке, на наш взгляд, удобнее всего решать программированием так как для перевода в троичную (и другие) систему счисления в электронных таблицах есть функция ОСНОВАНИЕ, которую выпускники с базовой подготовкой могут и не знать, могут возникнуть сложности с определением верхней границы диапазона исходного числа. Приведем пример решения этого задания с помощью программы⁴:

⁴ Здесь и далее все программы написаны на языке программирования Python

```

File Edit Format Run Options Window Help
def f(n):
    s=''
    while n>0:
        c=n%3
        s=str(c)+s
        n=n//3
    return s
for n in range(1,1000):
    n3=f(n)
    if n%3==0:
        n3=n3+n3[-2:]
    else:
        ost=n%3
        n3=n3+str(f(ost*5))
    r=int(n3,3)
    if r>=242:
        print(n)
        break

```

В этой программе сначала написана функция для перевода числа в троичную систему счисления, так как стандартной функции для этого в языке нет. Затем в цикле последовательно перебираются натуральные числа, для которых выполняется описанный в условии задачи алгоритм. Как только будет найдено первое натуральное число, удовлетворяющее условию задачи, оно выводится на печать и цикл досрочно прекращает работу.

Ответ: 11

Такой ответ дали 29,1% от всех, писавших этот вариант, что, конечно же, мало.

Задание 6 (содержательный раздел «Алгоритмы и программирование», средний процент выполнения 37,3%).

Пример формулировки:

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 t** (где t – целое число), вызывающая изменение направления движения на t градусов по часовой стрелке, **Налево t** (где t – целое число), вызывающая изменение направления движения на t градусов против часовой стрелки.

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

Черепаше был дан для исполнения следующий алгоритм.

Повтори 3 [Вперёд 33 Направо 90 Вперёд 41 Направо 90]

Поднять хвост

Вперёд 27 Направо 90 Вперёд 19 Налево 90

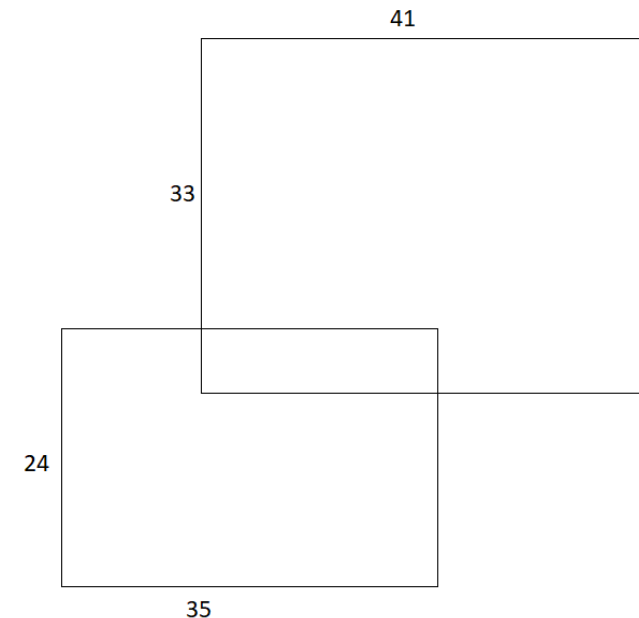
Опустить хвост

Повтори 3 [Вперёд 24 Направо 90 Вперёд 35 Направо 90]

Определите, сколько точек с целочисленными координатами находят внутри объединения фигур, ограниченного заданными алгоритмом линиями, включая точки на линиях.

Для решения этой задачи необходимо так же, как и в предыдущей, уметь выполнять алгоритм и уметь анализировать результат работы алгоритма. Только в случае задачи 6 алгоритм уже не линейный, а циклический и есть конкретный исполнитель. Возможно, из-за понятных действий исполнителя процент участников экзамена, которые не приступили к выполнению этой задачи, меньше, чем в задаче 5, и составляет 10,9%. Из-за того, что в 2025 году из регионального перечня программного обеспечения убрали школьный алгоритмический язык и среду программирования Кумир, решать это задание можно аналитически, выполнив алгоритм на черновике или в графическом редакторе. Также можно написать программу, которая будет строить фигуры, а затем выполнить подсчет точек. Наиболее быстрым является аналитический способ. Если проанализировать приведенный алгоритм, то можно увидеть, что результатом его работы является рисунок, изображенный справа.

Хорошо, если экзаменуемые будут уметь выполнять действия исполнителя, то есть смогут получить нужную фигуру без



использования вспомогательных инструментов. Во-первых, тем самым они демонстрируют свое умение выполнять действие циклического алгоритма, во-вторых, это самый быстрый способ для этой задачи, то есть он отлично подходит для всех групп подготовки. Для групп 1 и 2 – из-за простоты, для групп 3 и 4 – из-за скорости. Причем, исходя из тех ответов, которые дают участники экзамена, видно, что с задачей построения фигур все справляются достаточно хорошо. Сложности возникают с анализом полученной фигуры. В частности, обучающиеся:

— путают понятия «объединение» и «пересечение» (например, в приведенном примере 50 экзаменуемых правильно нашли количество точек для пересечения, и 63 – для объединения, как того требовалось в задаче);

— неправильно считают точки на линии фигур (по заданию они могут включаться или не включаться в подсчет, а экзаменуемые при подсчете не придают этому значение), но эта ошибка встречается уже реже.

Правильный ответ на это задание: 2167. Такой ответ дали 26,3% всех выпускников, писавших данный вариант. Конечно, успешнее всего с этим заданием справились выпускники, попавшие в группы 3 и 4. Вполне возможно, что немалая их часть для решения задачи воспользовалась программным способом, например, используя команды библиотеки **turtle** языка программирования Python. Также есть алгоритмы для автоматического подсчета точек, выпускники с высоким уровнем подготовки могли их применить.

Задание 8 (содержательный раздел «Теоретические основы информатики», средний процент выполнения 36,0%).

Пример формулировки:

Все шестибуквенные слова, составленные из букв Т, Е, О, Р, И, Я, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. EEEEEЕ
2. EEEEEИ
3. EEEEEО
4. EEEEEР
5. EEEEEТ
6. EEEEEЯ

.....

Определите, под каким номером в этом списке стоит последнее слово с чётным номером, которое не начинается с букв Е, И или О и при этом содержит в своей записи ровно одну букву Я.

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

Для решения таких задач все чаще выпускники применяют программирование, хотя возможен и другой способ решения этого задания: аналитический, с использованием систем счисления.

В открытом варианте этого года можно увидеть изменение, которое отличает его от привычных типовых формулировок: если в типовых задачах часто нужно искать просто номер слова, удовлетворяющего условию задачи, то в этой формулировке номер обязательно должен быть четным.

Решение задачи программным способом.

Для генерации слов воспользуемся функцией **product** из библиотеки **itertools**. Эта функция генерирует наборы символов заданного алфавита и заданной длины с учетом того, что символы могут повторяться. После генерации текущего слова вычисляется его номер и проверяются все условия, приведенные в задаче. Если эти условия выполняются, то номер сгенерированного слова выводится на печать. Так как по условию задачи необходимо найти номер последнего слова, то в качестве ответа приводится последнее выведенное число. Пример программы:

```
File Edit Format Run Options Window Help
from itertools import *
num=0
r=[]
for i in product('еиортя', repeat=6):
    num+=1
    if i[0] not in 'еио' and i.count('я')==1 and num%2==0:
        r.append(num)
print(r[-1])
```

В этой программе для увеличения скорости работы вывод всех номеров на экран заменен на запись их в список и вывод последнего элемента списка на печать.

Ответ: 45100. Такой ответ дали 33,6% из всех писавших этот

вариант.

Сложности и ошибки, возникшие при решении задания программным способом.

1. Не было учтено, что буквы слова должны быть выстроены в алфавитном порядке. Поэтому можно использовать команду **sorted**, чтобы не ошибиться при расстановке букв в алфавитном порядке или обратить на это особое внимание.

2. Многие участники экзамена, как всегда, зачастую находят то, что привыкли, а не то, что требуется в условии задачи, а именно количество комбинаций, удовлетворяющих условию. Также были даны ответы 23334 (номер первого, а не последнего слова, удовлетворяющего условиям задачи), 45101 (нечетный номер слова, подходящего под все остальные условия) и т.д.

3. Еще часто при решении задачи 8 (не в конкретном случае) экзаменуемые неверно выбирают команду, которую надо использовать. То есть если надо, чтобы буквы в комбинации не повторялись, лучше использовать команду **permutations()**, а если при этом надо, чтобы буквы в комбинации не переставлялись, необходимо использовать команду **combinations()**.

4. Некоторые экзаменуемые не могут правильно сформулировать условия для отбора нужных комбинаций или просто не знают, как задать некоторые условия.

Соответственно, для групп подготовки 3 и 4, кто выполняет задание 8, используя языки программирования, стоит порекомендовать для успешного его выполнения наreshивать большое количество разнообразных заданий, чтобы справляться с заданием быстро и без ошибок.

Решение задачи аналитическим способом.

Сначала расставим буквы в алфавитном порядке, получится последовательность ЕИОРТЯ. Теперь запишем последнее слово, которое не начинается на буквы ЕИО и содержит только одну букву Я. Очевидно, это слово должно начинаться с буквы Я, иначе оно не будет последним. По этой же причине все остальные позиции в искомом слове должна занять буква Т. Следовательно, под все условия, описанные в задаче, подходит слово ЯТТТТТ.

Теперь заменим каждую букву в алфавите на цифру: Е – 0, И – 1, О – 2, Р – 3, Т – 4, Я – 5. Перепишем найденное слово, заменив буквы на цифры, получим число 544444. Это число записано в шестеричной системе счисления, и для поиска номера этого числа необходимо перевести его в десятичную систему и не забыть прибавить единицу, так как номер слова в списке на единицу больше его десятичного кода.

Выполнив перевод, получим десятичный код слова 45100, его номер – 45101. Номер нечетный, поэтому берем предыдущее слово из списка, его номер 45100.

Сложности и ошибки, возникшие при решении задания аналитическим способом.

При решении задачи аналитическим способом традиционно допускаются ошибки при переборе вариантов элементов, в данной задаче с выбором комбинации, удовлетворяющей условию, и другие. Надо заметить, что процент участников экзамена, не дававших ответ на задачу 8, равен в среднем по области 15,5%, а конкретно для этого варианта он составляет 16,4%. Причем участники из групп 1 и 2 практически не справляются с этим заданием. Поэтому необходимо усилить базовую подготовку обучающихся по теоретическим основам информатики.

Задание 9 (содержательный раздел «Информационные технологии», средний процент выполнения 18,8%).

Пример формулировки:

Откройте файл электронной таблицы, содержащей в каждой строке шесть натуральных чисел. Определите среднее арифметическое чисел всех строк таблицы, для которых выполнены оба условия:

- в строке есть только одно число, которое повторяется дважды, остальные четыре числа различны;*
- повторяющееся число строки больше, чем среднее арифметическое четырёх её неповторяющихся чисел.*

В ответе запишите целую часть полученного числа.

Формулировка этого задания в текущем году немного изменилась в части постановки вопроса. Если во все предыдущие годы нужно было найти количество строк в таблице, которые удовлетворяют приведенным в задании условиям, то здесь необходимо вычислить среднее арифметическое всех чисел, содержащихся в строчках, подходящих под условия. Те учащиеся, которые научились решать только типовые задачи, не смогли дать правильный ответ. Этим можно объяснить такое резкое падение успешности выполнения этого задания, тем более, если в других вариантах также преобладали подобные формулировки.

Решать это задание можно с помощью электронных таблиц или программированием. Рассмотрим оба этих способа.

Решение с помощью электронных таблиц.

1. Определим, сколько раз в строке встречается каждое число (показана только первая строка файла):

Формула:

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	25	169	51	25	71	82	=СЧЁТЕСЛИ(A2:F2;A2)	=СЧЁТЕСЛИ(A2:F2;B2)	=СЧЁТЕСЛИ(A2:F2;C2)	=СЧЁТЕСЛИ(A2:F2;D2)	=СЧЁТЕСЛИ(A2:F2;E2)	=СЧЁТЕСЛИ(A2:F2;F2)

Результат:

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	25	169	51	25	71	82	2	1	1	2	1	1

2. Определим строки, которые удовлетворяют первому условию:

Формула:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2	25	169	51	25	71	82	=СЧ=СЧ=СЧ=СЧ=СЧ=СЧ	=И(СЧЁТЕСЛИ(G2:L2;2)=2;СЧЁТЕСЛИ(G2:L2;1)=4)					

Результат:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2	25	169	51	25	71	82	2	1	2	1	1		ИСТИНА

3. Находим среднее арифметическое неповторяющихся чисел:

Формула:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2	25	169	51	25	71	82	=СЧ	=СЧ	=СЧ	=СЧ	=СЧ	=СЧ	=И(СЧЁТЕСЛИ(G2:L2;1)=4;СРЗНАЧЕСЛИ(G2:L2;1;A2:F2))	

Результат:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2	25	169	51	25	71	82	2	1	2	1	1	ИСТИНА	93,25	

4. Находим значение повторяющегося числа:

Формула:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2	25	169	51	25	71	82	=СЧ	=СЧ	=СЧ	=СЧ	=СЧ	=СЧ	=И(СЧЁТЕСЛИ(G2:L2;2)=СРЗНАЧЕСЛИ(G2:L2;2;A2:F2))		

Результат:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2	25	169	51	25	71	82	2	1	2	1	1	ИСТИНА	93,25		25

5. Определяем строки, которые удовлетворяют второму условию:

Формула:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2	25	169	51	25	71	82	=СЧ	=СЧ	=СЧ	=СЧ	=СЧ	=СЧ	=И(СЧЁТЕСЛИ(G2:L2;"ИСТИНА")>2)	=СРЗНАЧЕСЛИ(I2:P2;"ИСТИНА";O2)	=СРЗНАЧЕСЛИ(I2:P2;"ЛОЖЬ";O2)	=O2>N2

Результат:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2	25	169	51	25	71	82	2	1	2	1	1	ИСТИНА	93,25	25		ЛОЖЬ

6. Далее с помощью фильтра выбираем те строки, для которых оба условия истинны:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
6	180	69	24	180	59	28	2	1	1	2	1	1	ИСТИНА	45	180	ИСТИНА
35	81	76	56	49	81	35	2	1	1	1	2	1	ИСТИНА	54	81	ИСТИНА
39	10	64	96	78	68	96	1	1	2	1	1	2	ИСТИНА	55	96	ИСТИНА
40	38	18	62	73	24	62	1	1	2	1	1	2	ИСТИНА	38,25	62	ИСТИНА
43	94	98	43	61	87	94	2	1	1	1	1	2	ИСТИНА	72,25	94	ИСТИНА

7. Выделяем диапазон с **исходными данными, попавшими в выборку** (столбцы A:F, их можно предварительно скопировать на другой лист рабочей книги). В правом нижнем углу строки состояния видим вычисленные для этого диапазона количество чисел, их сумму и среднее арифметическое:

Среднее: 58,96413502	Количество: 5214	Сумма: 307439
----------------------	------------------	---------------

8. Записываем в ответ только целую часть найденного среднего арифметического. Ответ: 58.

Сложности и ошибки, возникшие при решении задания с помощью электронных таблиц.

Наибольшая сложность при решении задачи 9 в выборе алгоритма решения. Большинство экзаменуемых, относящихся к группам 3 и 4, знают стандартный набор формул, которые могут использоваться при решении задач, знают, как пользоваться фильтрацией и сортировкой, но не всегда могут сформулировать правильно, какие конкретно формулы и в какой последовательности применить, чтобы учесть все условия.

Сложность задания для экзаменуемых групп 1 и 2 получается слишком высокой, и они либо в принципе не решают эту задачу, либо решают ее наугад, то есть применяют те формулы, которые знают, надеясь случайно получить правильный результат. Третья часть выпускников, писавших этот вариант, вообще не приступили к его выполнению. Ни один обучающийся из группы 1 с этим заданием не справился, а во второй группе процент успешности составил всего 4,9%.

Соответственно, чтобы повысить результат выполнения задания 9, необходимо опять же усилить базовую подготовку обучаемых в области использования электронных таблиц. Для этого в федеральной рабочей программе базового уровня предусмотрено 10 часов в 9 м классе и 6 часов в 10-м классе.

Выпускники, попавшие в группы 3 и 4, справились с этим заданием тоже хуже, чем в прошлом году.

Эту же задачу можно решить с помощью программы. Скорее всего, учащиеся с высоким уровнем подготовки использовали именно этот способ. При хорошем навыке программирования и понимании алгоритма на решение этой задачи тратится меньше времени.

Решение с помощью программирования.

Для начала следует подготовить файл с исходными данными. Для этого содержимое исходной электронной таблицы копируется и вставляется в текстовый документ, который затем открывается в программе. После открытия файла происходит построчное считывание данных и формирование списка, содержащего одну текущую строку. Этот список разбивается на два: для повторяющихся и неповторяющихся чисел. Затем происходит анализ всех условий из

```
File Edit Format Run Options Window Help
f=open('9.txt')
su=0
k=0
for s in f:
    a=[int(x) for x in s.split()]
    p=[x for x in a if a.count(x)==2]
    n=[x for x in a if a.count(x)==1]
    if len(p)==2 and len(n)==4 and p[0]>(sum(n)/4):
        su=su+sum(a)
        k=k+6
print(su/k)
```

формулировки задачи и вычисляется сумма и количество чисел.

В конце выводится среднее арифметическое. Пример программы:

Здесь **a** – список, содержащий числа из текущей строки, прочитанной из файла; **p** – список, содержащий повторяющиеся числа из строки; **n** – список, содержащий неповторяющиеся числа. Переменные **su** и **k** содержат сумму всех чисел из подходящих по условию строк, а также их количество. К переменной **su** каждый раз добавляется значение суммы элементов в исходной строке, а переменная **k** увеличивается на 6 так как в исходной строке 6 чисел.

После вывода на печать среднего арифметического следует взять только целую часть от найденного числа.

Ответ: 58.

Сложности и ошибки, возникшие при решении задания с помощью программы.

Основная сложность при таком способе решения, особенно, если выпускник уверенно решает типовые задачи с помощью программы, состоит в правильном подсчете суммы и количества чисел. Например, были даны ответы 353 (количество чисел увеличивалось не на 6, а на 1, т.е. неверно вычислялось количество чисел, удовлетворяющих условию), 869 (количество строк, удовлетворяющих условию как в типовых задачах) и другие неверные ответы с индивидуальными ошибками. Кстати, ответ 869 дали 22 выпускника, решавших этот вариант, а ответ 353 дали 9 человек, тогда как верный ответ дали 30 человек. То есть, если бы выпускники читали условие задачи более внимательно, количество правильно выполнивших это задание могло бы увеличиться вдвое.

Теперь рассмотрим задания повышенного уровня сложности.

Задание 11 (содержательный раздел «Теоретические основы информатики», средний процент выполнения 21,4%).

Пример формулировки:

На предприятии каждой изготовленной детали присваивают серийный номер, состоящий из 196 символов. В базе данных каждый серийный номер занимает одинаковое и минимально возможное число байт. При этом используется посимвольное кодирование серийных номеров, все символы кодируются одинаковым и минимально возможным числом бит. Известно, что для хранения 564 895 серийных номеров потребовалось не менее 145 Мбайт памяти. Определите минимально возможную мощность алфавита, используемого для записи серийных номеров. В ответе запишите только целое число.

Задачи подобного плана часто вызывают затруднения из-за проблем с анализом условия, вычислениями над большими числами и ограничениями по памяти, которые нужно обязательно учитывать. В условии задачи речь может идти как о максимальном размере памяти, выделенным для хранения информации так и о минимальном размере. От этого зависят особенности решения. В экзаменах предыдущих лет речь часто шла о максимальном размере файла, поэтому все дробные величины следовало округлять в меньшую сторону. Если этот подход применить к задаче в подобной формулировке, то будет получен неверный ответ. Те выпускники, которые недостаточно внимательно прочли условие задания, не очень хорошо усвоили материал темы «Измерение информации», решали задачу неправильно.

Посмотрим, какие темы, предусмотренные в федеральной рабочей программе ООО для 7–9-х классов базового и углубленного уровня и в федеральной рабочей программе СОО для 10–11-х классов базового и углубленного уровня, необходимо пройти для того, чтобы обучающиеся готовы были выполнить это задание:

7-й класс: «Представление информации», 9 часов (базовый, углубленный уровень);

10-й класс: «Представление информации в компьютере», 8 часов (базовый уровень); 19 часов (углубленный уровень);

11-й класс: «Информация и информационные процессы», 10 часов (углубленный уровень).

Из-за достаточно большого разрыва между изучением этих тем в основной и средней ступени часть обучающихся забывает изученный материал и решает задачи этого раздела неверно.

Рассмотрим решение вышеприведенной задачи.

1. Так как нам известен минимальный размер файла в Мегабайтах и количество серийных номеров, а также то, что для хранения одного серийного номера используется одинаковое и минимально возможное целое число байт, мы можем вычислить объем памяти в байтах, необходимый для хранения одного серийного номера:

<pre>>>> Type "help", "copyri 145*2**20/564895 269.15359491586935 >>> </pre>	Так как количество байт на один номер должно быть целым числом, следует полученное значение правильно округлить. Если мы округлим его до 269, то объем памяти для хранения всех номеров не превысит 145 Мб (он составит 144,9 Мб). Это недопустимо по условию задачи. Следовательно, эту величину нужно округлить до 270.
---	--

2. Зная количество байт, необходимых для хранения одного серийного номера, длину серийного номера и то, что для кодирования одного символа отводится одинаковое и минимально возможное целое значение бит, можно вычислить количество бит для кодирования одного символа из серийного номера:

```
>>> Type "help", "copyright", "credits" and run help()
>>> 270*8/196
11.020408163265307
>>> |
```

И вновь получается дробное число. Исходя из условий задачи, следует округлить эту величину до 11 бит на символ, так как это является минимально возможным количеством бит, не противоречащим прочим условиям задачи. До 12 бит это число округлять нельзя, так как это не минимальное количество бит.

Те выпускники, которые невнимательно, «по диагонали», прочитали условие задачи, сразу дают ответ 2048 символов в алфавите. Таких из тех, кто писал открытый вариант, оказалось 23 человека (из 215 приступивших к выполнению этого задания). На самом деле решение следующее:

3. Так как по условию задачи требуется найти минимально возможную мощность алфавита, она должна вмещаться в 11 бит и быть минимальной. Следовательно, это число 1025 – это минимальное число, которое поместится в 11 бит.

Ответ: 1025.

Сложности и ошибки, возникшие при решении задания.

1. Решая это задание неверно, выпускники дали несколько очень характерных ответов:
 - a. 2048 – максимально возможная мощность алфавита при длине кода 11 бит
 - b. 11 – длина кода вместо мощности алфавита
 - c. 2049 – округлили длину кода до 12 бит и взяли минимальное число, которое не войдет в этот код
 - d. 4096 – максимальная мощность алфавита для длины кода в 12 бит
 - e. Также было дано множество других неверных ответов, иллюстрирующих недостаточно сформированный навык смыслового чтения.

2. Задание 11, так же, как и задание 7, часто выполняется неверно из-за недостаточно хорошо усвоенной темы, связанной с кодированием и измерением информации. При изучении этой темы обучающиеся

часто просто зазубривают формулы, не вникая в их суть и не понимая смысла всех обозначений, которые есть в этих формулах. Из всех писавших открытый вариант 11,9% вообще не приступили к выполнению этого задания, по всей вероятности, так и не разобравшись в условии задачи и посчитав ее слишком сложной.

Задание 12 (содержательный раздел «Алгоритмы и программирование», средний процент выполнения 32,5%).

Это задание включено в экзамен с момента бумажной версии, перешло в компьютерную форму и вполне успешно решалось выпускниками предыдущих лет. В 2025 году успешность выполнения этого задания резко упала (от 57,5% в 2024 году до 32,5%). При этом формулировка задачи практически не изменилась по сравнению с задачами прошлых лет. Тем большее недоумение вызывают результаты его выполнения в этом году. Напомним эту задачу и вновь рассмотрим ее решение.

Пример формулировки:

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

А) заменить (v , w).

*Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Например, выполнение команды **заменить (111, 27)** преобразует строку 05111150 в строку 0527150. Если в строке нет вхождений цепочки v , то выполнение команды **заменить (v , w)** не меняет эту строку.*

Б) нашлось (v).

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА условие

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ условие ТО

команда1

ИНАЧЕ

команда2

КОНЕЦ ЕСЛИ

выполняется команда1 (если условие истинно) или команда2 (если условие ложно).

Дана программа для Редактора:

НАЧАЛО

ПОКА нашлось (52) ИЛИ нашлось (1122) ИЛИ нашлось (2222)

ЕСЛИ нашлось (52) ТО

заменить (52, 11)

КОНЕЦ ЕСЛИ

ЕСЛИ нашлось (2222) ТО

заменить (2222, 5)

КОНЕЦ ЕСЛИ

ЕСЛИ нашлось (1122) ТО

заменить (1122, 25)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

На вход приведённой выше программе поступает строка, начинающаяся с цифры «5», а затем содержащая n цифр «2» ($3 < n < 10\,000$). Определите наименьшее значение n , при котором сумма цифр в строке, получившейся в результате выполнения программы, равна 37.

Проще всего решить эту задачу, переписав приведенный алгоритм на язык программирования. Так как исходная строка не известна, ее нужно формировать в цикле, перебирая все допустимые значения количества цифр «2». Для каждой вновь сформированной строки необходимо выполнить приведенный алгоритм, затем найти сумму цифр в строке-результате. Если эта сумма окажется равной 37, нужно вывести текущее значение количества цифр «2» и прервать цикл.

Ответ: 78.

К выполнению этого задания не приступали 21,7% всех писавших открытый вариант. С ним не справились большое количество выпускников, попавших в группу 1 и 2. Так были даны очень характерные неверные ответы:

- строка 111111111111111111111111111111125 – результирующая строка для правильно найденного значения n.
- число 29 – значение переменной n, которое получается, если в параметрах метода **.replace** забыть указать количество замен за один шаг цикла;
- строка 115555555 – результирующая строка, которая получается при ошибке, описанной выше.

Для успешного выполнения этого задания выпускники должны хорошо владеть приемами работы со строками. Эта тема изучается в:

8-й класс: весь раздел «Алгоритмы и программирование», 21 час на базовом уровне и 34 часа на углубленном уровне (закладываются основы алгоритмизации, без которых невозможно дальнейшее изучение раздела);

9-й класс: «Разработка алгоритмов и программ», 6 часов на базовом уровне и 24 часа на углубленном уровне;

10-й класс: «Алгоритмы обработки символьных данных», 5 часов на углубленном уровне;

11-й класс: «Алгоритмы и элементы программирования», 11 часов на базовом уровне;

11-й класс: «Алгоритмы и структуры данных», 28 часов на углубленном уровне.

```
File Edit Format Run Options Window Help
for n in range(4,10000):
    s='5'+ '2'*n
    while '52' in s or '1122' in s or '2222' in s:
        s=s.replace('52','11',1)
        s=s.replace('2222','5',1)
        s=s.replace('1122','25',1)
    su=0
    for x in s:
        su=su+int(x)
    if su==37:
        print(n)
        break
```

С заданием 12 при должном вполне могут справляться и выпускники, имеющие средний уровень подготовки так как на данный момент формулировка этого задания не сильно отличается от множества задач из предыдущих экзаменов.

Задание 17 (содержательный раздел «Алгоритмы и программирование», средний процент выполнения 15,4%).

Процент выполнения этого задания совсем чуть-чуть превышает нижнюю границу допустимых значений. Это задание пока выполняется не очень хорошо, особенно выпускниками, попавшими в группы 1 и 2.

Пример формулировки:

В файле содержится последовательность целых чисел. Её элементы могут принимать целые значения от –100 000 до 100 000 включительно. Определите количество троек элементов последовательности, в которых только одно из чисел является трёхзначным, а сумма элементов тройки меньше максимального элемента последовательности, оканчивающегося на 17. В ответе запишите количество найденных троек чисел, затем максимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Задачу можно решать так же, как и задачу 9, и с использованием электронных таблиц, и написав программу. Формулировка и решение задачи 17 не очень изменились с прошлого года, выделим шаги решения задачи (рассмотрим один из возможных вариантов), которые можно реализовать как решая задачу в электронных таблицах, так и с помощью программы.

Шаг 1. Найти максимальный элемент последовательности, оканчивающийся на 17.

Шаг 2. Организовать перебор троек чисел, определить для каждой тройки условие трехзначности хотя бы для одного числа из нее.

Шаг 3. Определить для текущей тройки, не является ли сумма элементов, составляющих тройку, меньшей, чем число, найденное на шаге 1.

Шаг 4. Если это так, то найти количество таких троек и максимальную сумму.

Пример программы:

```

File Edit Format Run Options Window Help
f=open('17.txt')
a=[int(x) for x in f]
r=[]
m17=-100000
for x in a:
    if abs(x)%100==17:
        m17=max(m17,x)
print(m17)
for i in range(len(a)-2):
    if ((100<=abs(a[i])<=999)+(100<=abs(a[i+1])<=999)+(100<=abs(a[i+2])<=999))==1:
        if a[i]+a[i+1]+a[i+2]<m17:
            r.append(a[i]+a[i+1]+a[i+2])
print(len(r),max(r))

```

Ответ: 2781 85899

Сложности и ошибки, возникшие при решении задания.

Сложности при решении задачи 17 как при решении с использованием электронных таблиц, так и при использовании языков программирования – в последовательном соблюдении всех условий задачи, определения правильного порядка выполняемых действий.

Для экзаменуемых из групп 1 и 2 проще понять, как решать задачу с использованием электронных таблиц, тем более если они уже готовились к решению задания 9.

Участники групп 3 и 4 чаще решают эту задачу с помощью написания программ, причем здесь используются для решения стандартные алгоритмы нахождения суммы, среднего, количества, максимального или минимального чисел, удовлетворяющих определенным условиям.

Перечислим основные сложности и ошибки, возникающие при решении задачи 17 с помощью программ:

1. Решение задачи 17 выполняется, как правило, в два прохода: необходимо сначала найти значение, с которым будут сравниваться все элементы, а потом сравнить все элементы последовательности. Выделение этих двух шагов, как правило, у многих вызывает затруднение, а для экзаменуемых 1-й и 2-й групп, хорошо

справляющихся только с базовыми задачами содержательных линий, является практически невыполнимой задачей, так как в большинстве случаев у них нет достаточных навыков программирования.

2. Задание оценивается максимальным баллом, равным 1, только при условии правильного ответа на два поставленных вопроса, причем второй формулируется только тогда, когда говорится, что надо записать в качестве правильного ответа. Такая постановка вопроса часто путает экзаменуемых. То есть надо обязательно сразу дочитывать задачу до конца, чтобы понимать все этапы решения.

3. Еще одна распространенная ошибка связана с определением заданного окончания числа у отрицательных чисел, так как если просто брать остаток от деления на 10, 100 и т. д., как у положительных чисел, это может привести к неверному ответу.

4. То же можно сказать про определения значности числа у отрицательных чисел. Зачастую экзаменуемые проверяют, является ли число, например, трехзначным, проверкой принадлежности диапазону $[100;999]$, что исключает из трехзначных чисел все отрицательные. Такая ошибка была сделана рядом экзаменуемых, которые в качестве ответа дали пару чисел 1682, 85899.

Очевидно, что основной причиной низкого процента выполнения задачи является недостаточная подготовка экзаменуемых в разработке многошаговых алгоритмов решения задач, недостаточные навыки программирования, неумение правильно тестировать полученный алгоритм, недостаточный кругозор при работе с числами, что не позволяет выделить возможные сложности при решении задачи.

Наконец, рассмотрим задания высокого уровня сложности.

Задание 24 (содержательный раздел «Алгоритмы и программирование», средний процент выполнения 3,5%).

К выполнению этого задания не приступили в среднем по области 68,7%, фактически 2/3 от всех писавших экзамен. Задачу из открытого варианта не стали выполнять 64,7%. Можно сделать вывод о том, что в этом варианте задача показалась чуть легче, чем в других.

Пример формулировки:

Текстовый файл состоит из десятичных цифр и заглавных букв латинского алфавита. Определите в прилагаемом файле последовательность из максимального количества идущих подряд символов, среди которых ровно 55 любых цифр, начинающуюся буквой N и при этом не содержащую других N, кроме первой.

В ответе запишите число – количество символов в найденной последовательности.

Для выполнения этого задания следует написать программу.

Подобные задачи встречались в экзаменах предыдущих лет, и те выпускники, которые серьезно готовились к экзамену и имеют хорошую предметную подготовку, с ней справились: в группе 4 процент успешности ее выполнения составил 36,8%.

Чтобы понять алгоритм решения, лучше всего взять пример строки, соответствующей условию, и попробовать найти требуемую последовательность.

Пусть подобной строкой будет 'XX67XXNXX294XX5XXXNNNXX34XXX4XXNXXX6X87XX7XXX8XXX', и пусть нам следует найти последовательность символов максимальной длины, среди которых ровно 3 цифры, и которая начинается с символа 'N'.

Действие	Результат
Шаг 1. Заменим в исходной последовательности все символы 'N' на последовательность ' N'.	S = 'XX67XX NXX294XX5XXX N N NXX34XXX4XX NXXX6X87XX7XXX8XXX'
Шаг 2. Сформируем список, разрезав исходную строку по пробелам. Если первая цепочка символов не начинается с N, ее нужно удалить из списка.	A = ['NXX294XX5XXX', 'N', 'N', 'NXX34XXX4XX', 'NXXX6X87XX7XXX8XXX']
Шаг 3. Анализируем каждый элемент списка и вычисляем количество цифр в нем. Возможные ситуации: 1. Количество цифр меньше трех – такие строки нас не интересуют. 2. Количество цифр равно трем – значит, после третьей по счету цифры в строке больше нет других цифр, и мы вычисляем длину этого элемента. 3. Количество цифр больше трех – находим и запоминаем позицию четвертой по счету цифры, вычисляем длину строки до нее.	A[0]: количество цифр равно 4, номер позиции четвертой цифры – 8. Длина строки, которая нам подходит, равна 8 ('NXX294XX5XXX'). A[1], A[2]: цифр нет A[3]: количество цифр равно 3. Длина строки, которая нам подходит, равна 11 ('NXX34XXX4XX'). A[4]: количество цифр равно 5. Четвертая по порядку цифра стоит в позиции 10. Длина строки, которая нам подходит, равна 10 ('NXXX6X87XX7XXX8XXX'). Находим максимальную длину последовательности. Для нашего примера она равна 11.

4. Находим максимальную длину найденных цепочек.	
--	--

Реализуем описанный алгоритм в виде программы.

В этой программе мы запоминаем позицию 56-й по порядку цифры. Все найденные длины подходящих цепочек заносим в список, чтобы затем вывести его максимальный элемент.

Ответ: 132.

Сложности и ошибки, возникшие при решении задания.

Правильный ответ на эту задачу дали 3,3% от всех писавших этот вариант. Кроме правильного ответа экзаменуемые дали множество других неправильных ответов, среди которых выделяется ответ 123. Этот результат получился у тех выпускников, которые не учли ситуацию, когда в слове больше 55 цифр и следует искать 56-ю цифру. Этот ответ дали 9 человек, причем правильный ответ на эту задачу дали всего 8 выпускников.

Анализируя результаты выполнения задач 24, приходим к выводу, что для экзаменуемых представляет наибольшую сложность разработать алгоритм решения задачи с некоторым новым условием.

Но при подготовке к решению этой задачи главное помнить, что по спецификации она решается в 10-20 строк (в зависимости от стиля программирования может быть немного больше или меньше). Значит, на экзамене не надо паниковать, если встречается новая формулировка, а просто аккуратно проанализировать условие и попробовать найти все сложные ситуации. Конечно, для решения этой задачи требуется хорошая подготовка по типу групп 3 и 4, нужна насмотренность, необходимо понимание понятия полноты алгоритма.

Задача 25 (содержательный раздел «Алгоритмы и программирование», средний процент выполнения 5,3%).

```
File Edit Format Run Options Window Help
f=open('24.txt')
rez=[]
s=f.readline()
s=s.replace('N',' N')
a=[x for x in s.split()]
if s[0]!='N':
    a.remove(a[0])
for c in a:
    k=0
    for i in range(len(c)):
        if c[i] in '0123456789':
            k+=1
            if k==56:
                p=i
    if k==55:
        rez.append(len(c))
    if k>55:
        rez.append(p)
print(max(rez))
```

К выполнению этого задания нее приступили 81,6% выпускников. Это – самый высокий показатель в области среди всех заданий.

Задачу из открытого варианта правильно сделали всего 1,6%, что значительно ниже среднеобластного показателя. Следовательно, эта задача оказалась гораздо сложнее, чем в других вариантах. Рассмотрим ее.

Пример формулировки:

Напишите программу, которая перебирает целые числа, большие 2 920 573, в порядке возрастания и ищет среди них числа, представленные в виде произведения ровно двух простых множителей, не обязательно различных, каждый из которых содержит в своей записи ровно одну цифру 9.

В ответе в первом столбце таблицы запишите первые 5 найденных чисел в порядке возрастания, а во втором столбце – для каждого числа наибольший из соответствующих им найденных множителей.

Количество строк в таблице для ответа избыточно.

Задачи на поиск делителей всегда вызывают затруднения при решении, так как анализируемые числа достаточно большие, и решить задачу простым перебором за разумное время не всегда удастся. Следовательно, надо искать эффективное решение. К тому же, в данной задаче найденные делители должны удовлетворять большому количеству условий: их должно быть всего два, оба делителя должны быть простыми числами и содержать каждый только одну цифру 9. Скорее всего, такое обилие условий отпугнуло выпускников с недостаточно хорошей предметной подготовкой (даже в группе 4 ее решили правильно только 15,8% человек).

Решать эту задачу можно по-разному. Мы предлагаем следующий способ ее решения:

Действие	Фрагмент кода	
Шаг 1. Написать логическую функцию для определения, является ли число, в нее переданное, простым. В ней переменная k выступает в роли маркера, флага, показывающего, есть ли у числа N на диапазоне от 2 до N-1 целые делители. Если к концу работы		File Edit Format Run Options Window Help <pre>def simple(n): k=0 for i in range(2,n): if n%i==0: k=1 break return k==0</pre>

цикла маркер не поменял своего значения, то число является простым и функция возвращает значение True, иначе она возвращает значение False.	
Шаг 2. Так как в задаче следует найти первые пять чисел, заводим счетчик и задаем начальное значение числа. Здесь очень важно задать правильное начальное значение переменной <i>x</i>. Так как в условии сказано, что число должно быть больше 2920573, то в качестве начального значения берем число на единицу больше.	<pre>cnt=0 x=2920574</pre>
Шаг 3. До тех пор, пока не найдены требуемые пять чисел, в цикле ищем для текущего значения <i>x</i> его делители на «левой половине» диапазона делителей (от 2 до $\sqrt{x}+1$). Такой диапазон необходим, чтобы уменьшить количество итераций цикла и сократить время работы программы. Если в этой части диапазона был найден всего один делитель, значит и во второй половине диапазона такой делитель всего один. Определяем, являются ли найденные делители простыми	<pre>while cnt<5: k=0 for i in range(2, round(x**0.5)+1): if x%i==0: k=k+1 a=i if k==1: if simple(a) and simple(x//a): if str(a).count('9')==1 and str(x//a).count('9')==1: print(x,x//a) cnt+=1 x+=1</pre>

числами, считаем, сколько в них цифр 9.

Если оба делителя будут удовлетворять всем условиям, выводим текущее число, наибольший делитель, и увеличиваем счетчик найденных чисел на единицу. Затем увеличиваем на единицу текущее число.

Вся программа целиком:

Ответ:

```
2920661 153719
2920681 1709
2920781 8369
2920847 19603
2920849 6971
```

Правильный ответ на эту задачу дали всего 4 человека из всех писавших этот вариант. Еще 5 человек начали выполнять вычисления с числа 2920573 и дали неверные ответы. К тому же эти же 5 человек ограничили диапазон делителей значением квадратного корня числа x , потеряв возможные делители, если число являлось полным квадратом.

File Edit Format Run Options Window Help

```
def simple(n):
    k=0
    for i in range(2,n):
        if n%i==0:
            k=1
            break
    return k==0

cnt=0
x=2920574
while cnt<5:
    k=0
    for i in range(2, round(x**0.5)+1):
        if x%i==0:
            k=k+1
            a=i
    if k==1:
        if simple(a) and simple(x//a):
            if str(a).count('9')==1 and str(x//a).count('9')==1:
                print(x, x//a)
                cnt+=1
    x+=1
```

Некоторые выпускники не обратили внимание на то, что нужно найти первые 5 чисел, и записали лишние найденные числа, полностью заполнив таблицу для ответов.

Задание 26 (содержательный раздел «Информационных технологии», средний процент выполнения 3,0%).

Пример формулировки:

В магазине для упаковки подарков есть N кубических коробок из материалов двух видов. Самой интересной считается упаковка подарка по принципу матрёшки – подарок упаковывается в одну из коробок, та, в свою очередь, в другую коробку и т.д. Одну коробку можно поместить в другую, если длина её стороны хотя бы на D единиц меньше длины стороны другой коробки, при этом любые две соседние коробки сделаны из разных материалов. Известны длины сторон и материал коробок, имеющих в наличии. Определите наибольшее количество коробок, которое можно использовать для упаковки одного подарка и максимально возможную длину стороны самой маленькой из этих коробок. Размер подарка позволяет поместить его в самую маленькую коробку.

Входные данные

В первой строке входного файла находятся два натуральных числа через пробел: N ($N < 100\,000$) – количество коробок и D ($D < 10\,000$) – минимальная допустимая разность длин двух соседних коробок в «матрёшке». Каждая из следующих N строк содержит два разделённых пробелом натуральных числа, каждое из которых не превышает 10 000: длину стороны и условное обозначение вида материала коробки (0 или 1).

Запишите в ответе два числа: сначала наибольшее количество коробок, подходящих для упаковки подарка «матрёшкой», затем максимально возможную длину стороны самой маленькой коробки.

Типовой пример организации данных во входном файле

6 3 43 1 41 0 39 0 38 1 26 0 24 1

Пример входного файла приведён для шести коробок и случая, когда минимальная допустимая разница между длинами сторон коробок, подходящих для упаковки «матрёшкой», составляет 3 единицы.

При таких исходных данных условию задачи удовлетворяют наборы коробок с длинами сторон 24, 39, 43 (материалы этих коробок – 1, 0, 1 соответственно) или 26, 38, 41 (материалы – 0, 1, 0 соответственно). Таким образом, количество коробок равно 3, а максимально возможная длина стороны самой маленькой коробки равна 26.

Типовой пример имеет иллюстративный характер. Для выполнения задания используйте данные из прилагаемых файлов.

Решение задачи.

Процент выполнения задания, конечно, очень низкий. Многие не приступают к решению этой задачи, потому что для ее решения нет готовых «рецептов». Надо вдуматься в условие, прочитать несколько раз, найти все «подводные камни» и уже потом последовательно найти то, что требуется в условии задачи. Эта задача интересна тем, что для ее решения можно сочетать инструменты: например, отсортировать данные в электронных таблицах, посмотреть, как они выглядят, и затем написать программу с использованием уже подготовленных данных, иногда можно сразу написать программу для ее решения, иногда можно решить задачу только в электронных таблицах, и получается наглядней и быстрее.

Как раз задачу из представленного варианта очень удобно решать в электронных таблицах. Задача получилась несложная и решается за несколько минут. Надо только внимательно ее прочитать и понять, что требуется найти.

Для решения выполним следующую последовательность действий:

1. Импортируем данные из текстового файла на лист электронной таблицы. Для этого в открытой рабочей книге необходимо выбрать меню *Данные – Из текста*, указать имя файла, из которого берутся данные, и вставить данные на лист. Ниже приведен фрагмент полученной таблицы:

	A	B	C
1	4435	57	
2	125	0	
3	1991	1	
4	3902	0	
5	498	1	
6	8445	1	
7	5147	1	

В первой строке таблицы находятся два числа: количество коробок и минимальная разность двух соседних коробок. Первое число можно удалить, а второе перенести за пределы таблицы. Как видно из описания входных данных, в первом столбце находятся размеры коробок, а во втором – материал, из которого они сделаны. Из этих данных нужно собрать цепочку коробок максимальной длины так, чтобы разность между коробками превышала величину 57, а материал чередовался от коробки к коробке.

2. Чтобы это можно было сделать, сначала следует отсортировать данные. Так как коробку самого большого размера нам все равно придется забрать, сортировку делаем по убыванию. Ниже приведен фрагмент таблицы после сортировки.

	A	B	C
1	размер коробки	материал	
2	10000	0	
3	10000	1	
4	9999	0	
5	9998	0	
6	9994	0	
7	9993	1	
8	9993	0	
9	9992	1	

После сортировки становится ясно, что в списке есть две коробки максимального размера (10000), но сделанные из разного материала (0 и 1). Следовательно, возможны две цепочки коробок: первая начинается с коробки размером 10000 из материала 0, а вторая из коробки такого же размера, но из материала 1. Для решения задачи необходимо построить две цепочки коробок, удовлетворяющих условию задачи, и выбрать цепочку максимальной длины. Если окажется, что обе цепочки имеют одинаковую длину, надо выбирать ту, в которой самая последняя, маленькая коробка имеет наибольший размер.

3. Начнем формировать первую цепочку. Для этого в ячейки C2 и D2 запишем размер первой, самой большой коробки, и номер материала. Остальные ячейки этих столбцов заполним формулами:

	A	B	C	D	E	F	G	H	I
1	размер коробки	материал							57
2	10000	0	10000	0					
3	10000	1	=ЕСЛИ(И(C2-A3>=\$I\$1;D2<>B3);A3;C2)	=ЕСЛИ(И(C2-A3>=\$I\$1;D2<>B3);B3;D2)					
4	9999	0	=ЕСЛИ(И(C3-A4>=\$I\$1;D3<>B4);A4;C3)	=ЕСЛИ(И(C3-A4>=\$I\$1;D3<>B4);B4;D3)					
5	9998	0	=ЕСЛИ(И(C4-A5>=\$I\$1;D4<>B5);A5;C4)	=ЕСЛИ(И(C4-A5>=\$I\$1;D4<>B5);B5;D4)					
6	9994	0	=ЕСЛИ(И(C5-A6>=\$I\$1;D5<>B6);A6;C5)	=ЕСЛИ(И(C5-A6>=\$I\$1;D5<>B6);B6;D5)					
7	9993	1	=ЕСЛИ(И(C6-A7>=\$I\$1;D6<>B7);A7;C6)	=ЕСЛИ(И(C6-A7>=\$I\$1;D6<>B7);B7;D6)					
8	9993	0	=ЕСЛИ(И(C7-A8>=\$I\$1;D7<>B8);A8;C7)	=ЕСЛИ(И(C7-A8>=\$I\$1;D7<>B8);B8;D7)					

Результат:

26	9960	1	10000	0
27	9956	1	10000	0
28	9955	0	10000	0
29	9952	1	10000	0
30	9947	1	10000	0
31	9940	0	10000	0
32	9939	1	9939	1
33	9938	1	9939	1
34	9934	0	9939	1
35	9932	1	9939	1
36	9929	0	9939	1

4. Далее следует найти количество коробок, которые образовали цепочку. Это можно сделать, скопировав **значения**, вычисленные по формулам, вставив их в свободное место или на другой лист, и удалив дубликаты с помощью меню *Данные – Удалить дубликаты*. Фрагмент найденной цепочки:

Длина этой цепочки составляет 166 коробок, а максимальная длина стороны самой

маленькой коробки равна 63:

165	128	0
166	63	1

5. Теперь те же самые действия нужно сделать для цепочки, которая начинается с коробки длиной 10000, сделанной из материала 1. После всех тех же манипуляций получится цепочка, фрагмент которой приведен ниже:

Формулы для расчета:

	A	B	C	D	E	F	G	H	I
1	размер	материал							57
2	10000	0		10000					
3	10000	1	=ЕСЛИ(И(E3-A4>=I\$1;F3<>B4);A4;E3)		10000	1			
4	9999	0	=ЕСЛИ(И(E3-A4>=I\$1;F3<>B4);A4;E3)						
5	9998	0	=ЕСЛИ(И(E4-A5>=I\$1;F4<>B5);A5;E4)						

Итоговая цепочка (фрагмент):

10000	0
9939	1
9876	0
9819	1
9747	0
9690	1
9632	0
9570	1
9512	0
9443	1

V	W	X	Y
1	10000	1	
2	9940	0	
3	9881	1	
4	9824	0	
5	9766	1	
6	9696	0	
7	9636	1	
8	9577	0	
9	9519	1	
10	9462	0	
11	9401	1	
12	9337	0	
13	9279	1	

Длина этой цепочки также составляет 166, а максимальная длина стороны самой маленькой коробки составляет 37.

6. Раз обе цепочки имеют одинаковую длину, ответ на первый вопрос 166. Для ответа на второй вопрос нужно взять большую из полученных длин, т.е. 63.

Ответ: 166, 63

Сложности и ошибки, возникшие при решении задания.

Заметим, из 244 человек, кто выполнял этот вариант, только 8 дали какой-то ответ на задание. Из них только 4 человека дали правильные ответы на оба вопроса, и получили 2 балла; еще 4 человека ответили правильно на первый вопрос, получив 1 балл. Не приступили к выполнению этого задания 188 человек (77,1%). Рассмотрим, какие прослеживаются ошибки.

Основной прослеживаемой ошибкой является неверный порядок сортировки. Остальные сложности и ошибки по большому счету не связаны с конкретной формулировкой. Эта задача не предполагает, что на экзамене будет какая-то стандартная формулировка и для решения можно будет применить более или менее уже известное решение с измененными параметрами.

1. Для решения каждой задачи необходимо не просто придумать алгоритм решения задачи, а найти идею, которую можно реализовать в алгоритм, что является непосильной задачей без достаточного опыта решения задач высокой сложности.

2. В условии задачи кроются подводные камни, не найдя которые можно получить лишь один правильный ответ, а иногда и ни одного.

3. В решении многих задач можно использовать такие методы, как, например, динамическое программирование, но экзаменуемые, не владея ими, начинают реализовывать переборный алгоритм, который не может дать ответ за время экзамена.

4. В спецификации сказано, что задание проверяет умение обрабатывать целочисленную информацию с использованием сортировки. То есть, как правило, решение начинается с какой-либо сортировки. Эта подсказка помогает найти идею решения, а многие забывают про это.

Задание 27 (содержательный раздел «Алгоритмы и программирование», средний процент выполнения 7,0%).

В течение предыдущих лет это задание вызывало самые серьезные затруднения у выпускников (в 2023 году процент выполнения составил 3,0%, в 2024 – 2,4%). В этом году с заданием 27 справились лучше. Оно также осталось заданием высокого уровня сложности, но из-за полностью изменившегося содержания задачи, на наш взгляд, ее стало решать немного проще. Рассмотрим эту задачу.

Пример формулировки.

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 этого кластера, сумма расстояний от которой до всех остальных точек кластера 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:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

В файле А хранятся координаты точек двух кластеров, где $H = 6$ и $W = 4,5$ для каждого кластера. В каждой строке записана информация о расположении на карте одной звезды: сначала координата x , затем координата y . Известно, что количество точек не превышает 1000.

В файле Б хранятся координаты точек трёх кластеров, где $H = 6$, $W = 5$ для каждого кластера. Известно, что количество точек не превышает 10 000. Структура хранения информации в файле Б аналогична структуре в файле А.

Известно, что в файле Б имеются координаты ровно трёх «лишних» точек, являющихся аномалиями, возникшими в результате помех при передаче данных. Эти три точки не относятся ни к одному из кластеров, их учитывать не нужно.

Для файла А определите координаты центра каждого кластера, затем найдите два числа: P_x – среднее арифметическое абсцисс центров кластеров, и P_y – среднее арифметическое ординат центров кластеров.

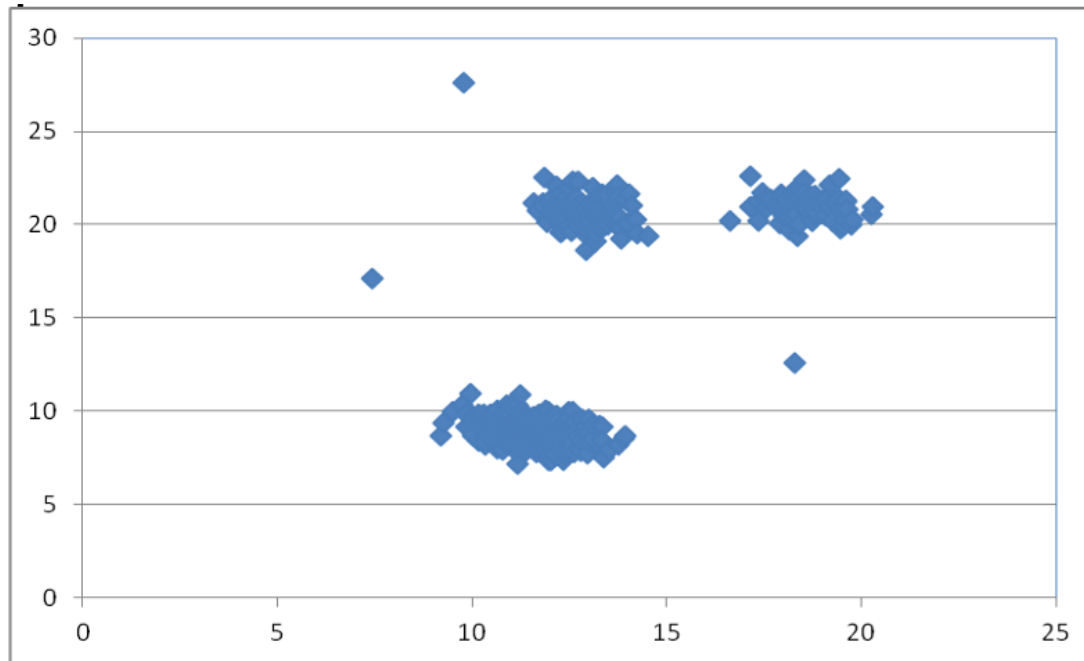
Для файла Б определите координаты центра каждого кластера, затем найдите два числа: Q_x – сумму абсцисс центров кластеров с минимальным и максимальным количеством точек, и Q_y – сумму ординат центров кластеров с минимальным и максимальным количеством точек.

Гарантируется, что во всех кластерах количество точек различно.

В ответе запишите четыре числа: в первой строке – сначала абсолютную величину целой части произведения $P_x \times 10\,000$, затем абсолютную величину целой части произведения $P_y \times 10\,000$; во второй строке – сначала абсолютную величину целой части произведения $Q_x \times 10\,000$, затем абсолютную величину целой части произведения $Q_y \times 10\,000$.

Возможные данные одного из файлов проиллюстрированы графиком.

Внимание! График приведён в иллюстративных целях для произвольных значений, не имеющих отношения к заданию. Для выполнения задания используйте данные из прилагаемого файла



Кроме привычных текстовых файлов к задаче приложены еще два файла в формате электронных таблиц.

Решение задачи.

Весь ход решения этой задачи можно разбить на три этапа:

1. Кластеризация – разделение всех данных по кластерам.
2. Визуализация – графическое изображение кластеров для контроля правильности выполнения первого этапа.
3. Поиск центров для каждого кластера и вычисление требуемых в задаче величин.

Рассмотрим каждый этап отдельно.

1. Кластеризация может быть выполнена различными методами. В этом решении мы использовали метод DBSCAN, адаптированный для решения задачи 27. Сущность этого метода заключается в том, что берется произвольная точка из приведенного набора и добавляется в новый кластер. Далее в массиве исходных данных находятся точки, лежащие на расстоянии не более некоторой величины **epsilon**. Они также добавляются в этот

кластер. В этот же кластер добавляются те точки, которые являются «соседями» для точек, уже находящихся в кластере. Все добавленные в кластер точки удаляются из исходного массива. И так продолжается до тех пор, пока в исходном массиве больше не окажется никаких точек. Если в массиве есть аномальные точки (в описании задачи для файла В об этом идет речь), они тоже образуют отдельные кластеры размером в одну точку каждый. Их можно просто игнорировать при поиске центров.

```
File Edit Format Run Options Window Help
from math import *
f=open('27_A.txt')
a=[]
#-----Кластеризация-----
clast=[]
epsilon=1
for s in f:
    s=s.replace(',','.')
    x,y=map(float,s.split())
    a.append([x,y])
while a:
    clast.append([a[0]])
    a.remove(a[0])
    for x in clast[-1]:
        for y in a[:]:
            if dist(x,y)<epsilon:
                clast[-1].append(y)
                a.remove(y)
```

определения, находятся ли две точки в одном кластере, используется стандартная функция **dist** из библиотеки **math**. Эта функция вычисляет декартово расстояние между двумя точками (причем одна точка уже находится в кластере, а вторая пока нет), и если это расстояние окажется не более **epsilon**, то вторая точка добавляется в текущий кластер и удаляется из исходного массива. После выполнения этого фрагмента программы будут созданы кластеры

Фрагмент программы, отвечающий за кластеризацию:

Здесь **a** – исходный список, прочитанный из файла, **clast** – список, содержащий найденные кластеры, **epsilon** – предельное расстояние, на котором должны находиться точки, попавшие в один кластер. Это расстояние выбирается эмпирически, значение можно корректировать, если после визуализации окажется, что точки по кластерам распределены неверно.

При формировании массива **a** следует учесть, что в файле первая строка содержит заголовок “X Y”. Его следует предварительно удалить (или прочитать в «никуда»).

После того, как прочитана строка с числовыми данными, в ней необходимо провести замену запятой на точку. Иначе при чтении в переменные преобразовать данные к типу **float** окажется невозможно.

После формирования исходного списка с точками начинается разбиение этих точек по кластерам. Для

произвольной формы, по которым распределены все точки из исходного файла. Правильно ли произошла кластеризация, сколько кластеров получилось можно проконтролировать на следующем этапе.

2. Визуализация. Для этого нужно предварительно открыть файл A, сохраненный в формате электронной таблицы, и построить по всем данным из него точечную диаграмму. Это позволит убедиться, что кластеров

```
#-----Визуализация-----
from turtle import *
screensize(2000,2000)
pu()
k=10
tracer(0)
colors=['red','green','blue','brown','pink']
for i in range(len(clast)):
    for x,y in clast[i]:
        goto(x*k,y*k)
        dot(5,colors[i])
update()
```

действительно два и увидеть, как они расположены. После этого в программу нужно добавить следующую часть:

Здесь происходит подключение стандартной библиотеки **turtle**, позволяющей рисовать. Задаются размеры полотна, на котором будет построено изображение, коэффициент масштабирования и пр.

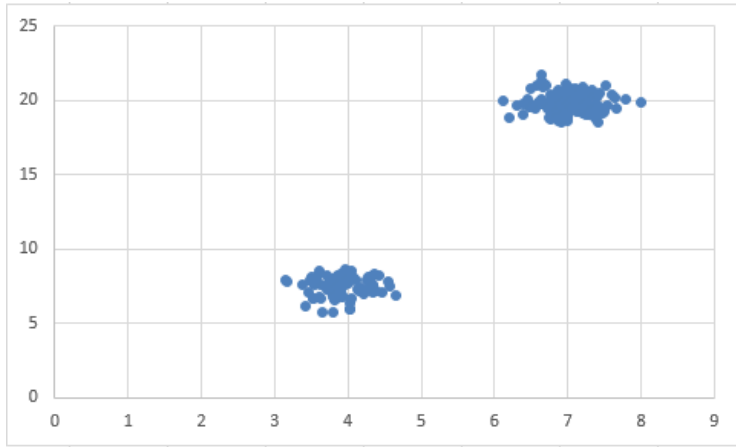
Также необходимо создать список с цветами, в которые будут окрашиваться точки в каждом кластере. Цветов лучше брать с избытком, это позволит найти

ошибку, если кластеров будет построено больше, чем указано в условии задачи.

После этого происходит переход в точку с координатами из кластеров и вывод точки на экран.

Если после визуализации окажется, что кластеров получилось не два, а больше (или меньше), следует вернуться в раздел кластеризации и изменить значение параметра **epsilon**.

Результаты визуализации:



Точечная диаграмма из электронной таблицы

Кластеризация выполнена верно. Можно переходить к третьему этапу.

3. Фрагмент программы для поиска центров и вычисления необходимых величин для файла А показан на рисунке справа.

Так как из предыдущих этапов известно, что кластеров действительно два, можно создать список **centr**, состоящий из двух элементов, куда затем будут помещены центры каждого кластера.

Поиск центра каждого кластера происходит с помощью полного перебора всех точек в кластере и вычисления суммарного расстояния от каждой точки до всех остальных точек кластера. Здесь величина **m_dist** означает минимальное суммарное расстояние, **s** – суммарное расстояние от текущей точки до всех остальных точек кластера.

После нахождения центров обоих кластеров следует вычисление величин R_x и R_y и вывод их на экран в требуемом виде.

Для файла А ответом будут числа 55295 и 136872.

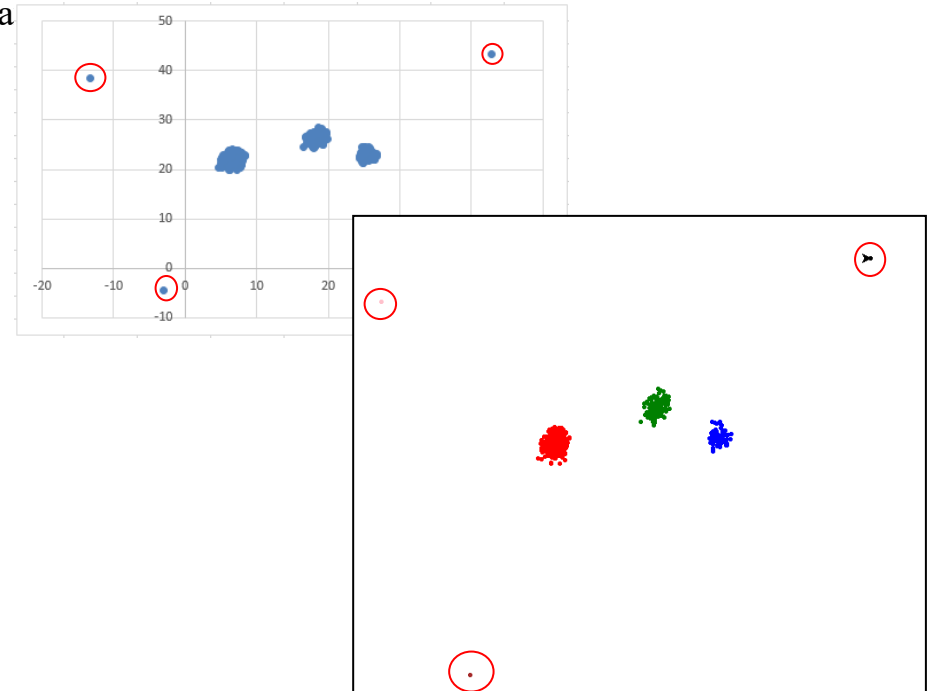
Результат работы модуля **turtle**

```
#-----Поиск центров-----
centr=[[ ],[ ]]
for i in range(len(clast)):
    m_dist=10**10
    for x in clast[i]:
        s=0
        for y in clast[i]:
            s=s+dist(x,y)
        if s<m_dist:
            m_dist=s
            centr[i]=x
print(centr)
px=(centr[0][0]+centr[1][0])/2*10000
py=(centr[0][1]+centr[1][1])/2*10000
print(abs(int(px)),abs(int(py)))
```

Для решения этой задачи с использованием файла В можно использовать точно такой же способ, ничего не меняя на этапах кластеризации и визуализации. Единственное, следует добавить в массив цветов еще 2-3 других цвета.

Часть программы для файла В и результаты визуализа

```
File Edit Format Run Options Window Help
from math import *
f=open('27_B.txt')
a=[]
#-----Кластеризация-----
clast=[]
epsilon=1
for s in f:
    s=s.replace(' ','.')
    x,y=map(float,s.split())
    a.append([x,y])
while a:
    clast.append([a[0]])
    a.remove(a[0])
    for x in clast[-1]:
        for y in a[:]:
            if dist(x,y)<epsilon:
                clast[-1].append(y)
                a.remove(y)
#-----Визуализация-----
from turtle import *
screensize(2000,2000)
pu()
k=10
tracer(0)
colors=['red','green','blue','brown','pink','black','yellow']
for i in range(len(clast)):
    for x,y in clast[i]:
        goto(x*k,y*k)
        dot(5,colors[i])
update()
```



Здесь красным цветом обведены аномалии, которые есть в данных файла В (точки коричневого, розового и черного цветов). Они видны как на диаграмме, построенной в электронных таблицах, так и на рисунке, нарисованном программой. Для дальнейшего решения следует взять только кластеры, в которых точки окрашены в красный, зеленый и синий цвета. Остальные кластеры можно из списка кластеров удалить или просто работать с **clast[0]**, **clast[1]** и **clast[2]**.

Так как в формулировке задачи для файла В есть отличия, третью часть программы (поиск центров) надо немного изменить. Для файла В необходимо вычислить сумму абсцисс центров кластеров с минимальным и максимальным количеством точек, а также сумму ординат центров кластеров с минимальным и максимальным количеством точек. После поиска центров следует найти такие кластеры. Например, можно сформировать список **dl**, куда записать длины кластеров:

```
#-----Поиск центров-----
centr=[[],[],[]]
for i in range(3):
    m_dist=10**10
    for x in clast[i]:
        s=0
        for y in clast[i]:
            s=s+dist(x,y)
        if s<m_dist:
            m_dist=s
            centr[i]=x
print(centr)
dl=[]
for i in range(3):
    dl.append(len(clast[i]))
print(dl)
```

Результат работы этого фрагмента:

[397, 137, 72]

Следовательно, для дальнейших вычислений нужно брать центры для кластеров с номерами 0 и 2.

Фрагмент программы для поиска центров и вывода найденных величин на печать:

```

#-----Поиск центров-----
centr=[[ ],[ ],[ ]]
for i in range(3):
    m_dist=10**10
    for x in clast[i]:
        s=0
        for y in clast[i]:
            s=s+dist(x,y)
        if s<m_dist:
            m_dist=s
            centr[i]=x
dl=[]
for i in range(3):
    dl.append(len(clast[i]))
print(dl)
qx=(centr[0][0]+centr[2][0])*10000
qy=(centr[0][1]+centr[2][1])*10000
print(abs(int(qx)), abs(int(qy)))

```

Ответ для файла В: 319184 448643

Конечно, возможны и другие способы решения этой задачи.

Правильные ответы на эту задачу дали только 12 человек, выполнявших этот вариант. Еще 10 человек дали правильный ответ только для файла А, так как для файла В они также нашли среднее арифметическое абсцисс и среднее арифметическое ординат всех центров. Очевидно, эти выпускники недостаточно внимательно прочли условие задачи или сделали так по привычке, решая в течение учебного года множество однотипных задач, где не было разницы между расчетами по файлу А и расчетами по файлу В. Также несколько человек получили 1 балл, найдя для файла В сумму абсцисс и ординат для центров всех кластеров, без учета количества точек в них. И это вновь приводит нас к проблеме невнимательности при чтении условия задачи и наreshивания типовых, шаблонных задач.

Так как задача появилась в экзамене впервые, в следующие годы можно ждать ее изменения и увеличения количества вариантов условий. Поэтому при изучении соответствующей темы на уроках информатики следует

уделить серьезное внимание в том числе и рассмотрению различных методов кластеризации данных: кроме метода DBSCAN существуют и другие методы разделения точек на кластеры, например, метод K-means.

1.1.3 Анализ метапредметных результатов обучения, повлиявших на выполнение заданий КИМ

Результаты выполнения заданий КИМ ЕГЭ по информатике в значительной мере определяются уровнем сформированности метапредметных результатов обучения. Результаты 2025 года показывают, что целый комплекс результатов сформирован на достаточном уровне, позволяющем выпускникам эффективно справляться с выполнением заданий.

Прежде всего, следует отметить, что учащиеся всех групп подготовки справлялись с линиями заданий на базовом уровне сложности, проявляя владение навыками познавательной деятельности. Так, подавляющее большинство заданий, в которых проверялось владение понятиями, выполнено выпускниками региона, что свидетельствует об освоении метапредметного результата, связанного с составляющей познавательной деятельности — формированием понятий на примере учебного предмета «Информатика».

Подтверждает достаточный уровень сформированности познавательной деятельности выполнение заданий, направленных на работу с информацией, на установление существенных признаков или оснований для сравнения, классификации и обобщения, на анализ и преобразование информации различных форм представления (задания 1-4, 13 и др. – 57,8% в среднем).

Средние показатели характеризует познавательный навык, который учащиеся продемонстрировали в заданиях на самостоятельный поиск решений практических задач, применение различных методов познания (задания 7, 8 и др. – 32,87% в среднем). Вместе с тем следует выделить метапредметные результаты, недостаточный уровень сформированности которых способствовал возникновению затруднений в выполнении экзаменационной работы. Прежде всего, следует отметить ошибки вследствие низкой читательской компетенции и невнимательного прочтения текста. Это определяется, во-первых, уровнем сформированности метапредметного результата обучения — умением ориентироваться в различных источниках информации, критически оценивать и интерпретировать информацию. В результате учащиеся выхватывают фрагменты информации, представленные в явном виде, не вчитываясь полностью в

содержание. Недостаточно ориентируясь в предоставляемой информации, учащиеся опирались на ожидаемые формулировки заданий, стремясь минимизировать усилия, решали задачи, отработанные на уроках. Наиболее ярко это выражено при решении задач 5, 6, 9 и др. (29% в среднем. Меньше всего процент выполнения – 18,8% задания 9). Значительный процент ошибок в экзаменационных работах определяется недостаточным уровнем умений базовых исследовательских и логические действия, смыслового чтения (навыка системного анализа данных и невнимательным прочтением условия задач). Особенно это ярко проявляется при решении заданий 24-27 (процент выполнения 4,7). Так как ЕГЭ по информатике носит практико-ориентированный характер, то задачи, выполняемые с помощью компьютера и программного обеспечения, показывают наибольшие проблемы формирования метапредметных результатов. Необходимо продолжить работу по формированию базовых исследовательских действий, умения построения логически верных рассуждений и подтверждения их истинности, умения анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях. Для решения задач 24–27 необходимо критическое мышление и креативный подход, который можно сформировать у выпускников только при системном подходе. Невысокие результаты вышеперечисленных заданий позволяет сделать вывод о наличии пробелов в формировании метапредметных результатов у обучающихся при изучении информатики. Анализ результатов ЕГЭ 2025 года по группам подтверждает выводы:

- Группа 81-100 баллов: демонстрирует довольно высокий уровень всех метапредметных навыков, успешно справляясь с заданиями базового и повышенного уровня, исключая сложные (задание 27 – 60%, в среднем – 43,75%).

- Группа 61-80 баллов: показывает хорошую базовую подготовку, но испытывает значительные трудности с заданиями, требующими творческого, нешаблонного подхода и самостоятельного написания программ (задание 24 – 2,7%, задание 25 – 6,5%).

- Группы ниже 60 баллов: обнаруживают довольно низкий уровень сформированности ключевых метапредметных результатов. Их результаты показывают, что они овладели лишь отдельными фрагментарными универсальными действиями, но испытывают затруднения с умением применять их комплексно для решения задач.

Таким образом, выпускники в целом овладели базовыми познавательными действиями (анализ, преобразование информации, работа с моделями), при этом наблюдается острый дефицит метапредметных умений высшего порядка: смысловое чтение, системный анализ, критическое и алгоритмическое мышление, планирование деятельности.

Задания практико-ориентированного характера, требующие самостоятельного конструирования решения (написание программ, анализ данных), являются наибольшим барьером. Это указывает на то, что обучение часто носит репродуктивный, а не деятельностный характер.

В следующей таблице приведем номера заданий, вызвавших сложности у участников экзамена, и выделим требования к метапредметным умениям из таблицы 1 кодификатора, недостаточная сформированность которых влияет на успешность выполнения указанных заданий.

Номера заданий	Код требований	Ошибки при выполнении заданий КИМ, обусловленные слабой сформированностью метапредметных умений
5, 6	1.1	Ошибки интерпретации алгоритмов, приведенных на естественном языке, которые ведут к неправильному результату работы. Затруднения, возникающие при усложнении алгоритма. При подготовке к экзамену обучающиеся не выделили существенные для выполнения задачи моменты. В результате либо находили не ту величину, которую требовалось, либо неверно выполняли вычисления.
	1.2	Участники экзамена не могут сформировать правильную последовательность действий в соответствии с условием задачи.
8	1.1.1	Участники экзамена не могут выделить все существенные признаки, чтобы правильно задать условие для решения задачи.
	1.2.1	Участники не используют или не могут применить знания, полученные на математике при решении этой задачи.
9	1.3	Экзаменуемые не всегда могут сформулировать правильно, какие конкретно формулы и в какой последовательности применить, чтобы учесть все условия. На выполнение этого задания зачастую ученики тратят больше времени, чем рассчитывали, некоторые из-за этого начинают паниковать и записывают ответ, не удостоверившись в его правильности.
11	1.2	Участники экзамена не могут сформировать правильную последовательность действий в соответствии с условием задачи. При получении промежуточного или окончательного результата экзаменуемые не проверяют его на достоверность (проверка правильности единиц измерения, соответствие полученного числового значения реальности, соответствие полученного значения условию задачи).
17	1.2	Экзаменуемым тяжело выстроить правильную последовательность действий для решения задачи.
	3.1, 3.2	Безошибочное выполнение задания требует от учеников хорошей теоретической подготовки по теме задания, знания широкого спектра алгоритмов. Участники экзамена не учитывают этого, не могут оценить степень готовности к выполнению этого задания, в результате при изменении формулировки не справляются с ним.
24 – 26	1.2	При недостаточной сформированности указанного умения экзаменуемые не могут найти идею решения в этих

		задачах.
	3.1, 3.2	На успешное выполнение этих заданий в большой степени влияет степень самоорганизации и самоконтроля обучающихся.
27	1.1	Участники экзамена не могут выстроить цепочку действий для решения задачи, анализировать полученные результаты для коррекции разработанного алгоритма

1.1.4 Выводы об итогах анализа выполнения заданий, групп заданий:

- *Перечень элементов содержания / умений и видов деятельности, усвоение которых всеми школьниками региона в целом можно считать достаточным*
 - Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы);
 - Умение строить таблицы истинности и логические схемы;
 - Умение поиска информации в реляционных базах данных;
 - Умение кодировать и декодировать информацию;
 - Информационный поиск средствами текстового процессора
- *Перечень элементов содержания / умений и видов деятельности, усвоение которых всеми школьниками региона в целом, школьниками с разным уровнем подготовки нельзя считать достаточным*
 - 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;
 - Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов;
 - Знание основных понятий и методов, используемых при измерении количества информации;
 - Умение обрабатывать числовую информацию в электронных таблицах;
 - Умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд;

- Умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования;
- Умение создавать собственные программы (10–20 строк) для обработки символьной и целочисленной информации;
- Умение обрабатывать целочисленную информацию с использованием сортировки;
- 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

○ *В дополнение к перечисленным в предыдущем пункте элементы содержания / умений и видов деятельности, усвоение которых участниками первой и второй групп нельзя считать достаточным:*

- Оперировать массивами данных;
- Умение исполнить рекурсивный алгоритм;
- Анализировать результат исполнения алгоритма;
- Анализировать текст программы с точки зрения соответствия записанного алгоритма поставленной задаче и изменять его в соответствии с заданием;
- Построить дерево игры по заданному алгоритму и обосновать выигрышную стратегию.

○ *Выводы об изменении успешности выполнения заданий разных лет по одной теме / проверяемому умению, виду деятельности*

С уверенностью можно сказать, что при отсутствии существенных изменений в заданиях результаты их выполнения повышаются. Это можно проследить, например, анализируя результаты выполнения заданий 2, 4, 18, 19, 20.

Во всех заданиях, где изменилось задание, изменилась формулировка вопроса, произошли любые изменения, произошло понижение процента выполнения. К сожалению, это свидетельствует о том, что большинство участников

экзамена имеют недостаточный уровень предметных результатов обучения, компенсируя это при подготовке к экзамену тренировкой выполнения конкретных заданий.

Отмечается, что выполнение многих заданий, не проверяющих знания и навыки по разделу алгоритмизации и программирования, решаются участниками экзамена этими методами (например, задания 8 или 15). Такие методы выбираются из-за кажущейся простоты и более гарантированного правильного результата. Но их использование не приводит к повышению результатов выполнения заданий, в лучшем случае результаты остаются на прежнем уровне. Это связано как с недостаточными умениями и знаниями, проверяемыми в этих заданиях по спецификации, так и с недостаточными навыками программирования. Тем не менее можно отметить улучшение показателей при выполнении некоторых заданий, что свидетельствует о повышении знаний и умений по некоторым разделам информатики. Перечислим такие задания и о совершенствовании каких умений они свидетельствуют:

Задание 1: Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)

Задание 3: Умение находить информацию в реляционных базах данных

Задание 7: Умение определять объём памяти, необходимый для хранения графической и звуковой информации;

Задание 15: Знание основных понятий и законов математической логики

○ *Выводы о связи динамики результатов проведения ЕГЭ с использованием рекомендаций для системы образования Иркутской области и системы мероприятий, включенных в статистико-аналитические отчеты о результатах ЕГЭ по учебному предмету в предыдущие 2-3 года.*

Решение участниками экзамена многих заданий методами программирования свидетельствует об эффективности проводимых ИРО курсов повышения квалификации для учителей Иркутской области «Методические и практические аспекты обучения по разделу "Алгоритмизация и основы программирования" предмета "Информатика"» в объеме 72 часов.

Разработка методических рекомендаций по преподаванию информатики в школах Иркутской приводит к исключению некоторых ошибок при выполнении отдельных заданий экзамена.

Региональный семинар для учителей информатики «Вопросы, возникшие при проведении ЕГЭ в компьютерной форме» позволяет расширить географию образовательных организаций, ученики из которых принимают участие в экзамене.

Региональная научно-практическая конференция «Опыт, проблемы и перспективы естественно-математического образования» позволяет находить методы совершенствования методики проведения занятий и делится опытом эффективного развития предметных, метапредметных и универсальных компетенций.

Раздел 2. РЕКОМЕНДАЦИИ ДЛЯ СИСТЕМЫ ОБРАЗОВАНИЯ ИРКУТСКОЙ ОБЛАСТИ

2.1. Рекомендации по совершенствованию организации и методики преподавания предмета в Иркутской области на основе выявленных типичных затруднений и ошибок

2.1.1....по совершенствованию преподавания учебного предмета всем обучающимся

○ Учителям

1 Спланировать работу с рекомендациями отчета для выявления дефицитов школьников в освоении федеральной рабочей программы по информатике на всех уровнях преподавания (НОО, ООО, СОО).

2 Сопоставить перечень (кодификатор) распределенных по классам проверяемых требований к результатам освоения основной образовательной программы НОО, ООО, СОО и элементов содержания по информатике с текущими результатами и результатами внешних оценочных процедур. На основе этого определить темы, по которым школьники конкретной группы показывают недостаточные знания и умения.

3 Спланировать систематическую работу на учебных занятиях по устранению типичных ошибок по предмету.

В качестве рекомендаций для предотвращения типичных ошибок при сдаче ЕГЭ предлагается:

- использовать технологии активного обучения, взаимообучения, технологию «Смешанного обучения»;
- уделять внимание развитию межпредметных связей, особенно связи информатики и математики;
- включать задания ЕГЭ при объяснении учебного материала, решении задач и выполнении практических работ по всем темам курса информатики;
- обращать внимание, прежде всего, на ключевые базовые темы, включённые в ЕГЭ.

Учителям начального общего образования необходимо включать в учебные и внеурочные задания на формирование следующих познавательных универсальных учебных действий:

- базовых логических действий: устанавливать связи и зависимости между математическими объектами («часть-целое», «причина-следствие»); применять базовые логические действия: сравнивать,

анализировать, классифицировать, обобщать; представлять текстовую задачу, её решение в виде модели, схемы;

- базовых исследовательских действий: применять изученные методы познания (измерение, моделирование, перебор вариантов);
- информационные действия: находить и использовать для решения учебных задач текстовую, графическую информацию в разных источниках информационной среды; читать, интерпретировать графически представленную информацию (схему, таблицу, диаграмму, другую модель);

На получение предметного результата включать задания:

- на чтение, запись, сравнение, упорядоченность;
- на выполнение арифметических действий;
- распознавание верных (истинных) и неверных (ложных) утверждений со словами: «все», «некоторые», «и», «каждый», «если..., то...»;
- формулировать утверждение (вывод), строить логические рассуждения;
- классифицировать объекты по одному-двум признакам.

Учителям на уровне основного общего образования:

При изучении курса информатики в 7 классе особое внимание требуется уделить следующим элементам содержания:

- Программное обеспечение компьютера. Прикладное программное обеспечение. Системное программное обеспечение. Системы программирования. Правовая охрана программ и данных.
- Информационные процессы – процессы, связанные с хранением, преобразованием и передачей данных. Информационный объём данных.
- При изучении курса информатики в 8 классе особое внимание требуется уделить следующим темам содержания:

- Разработка для формального исполнителя алгоритма, приводящего к требуемому результату при конкретных исходных данных. Разработка несложных алгоритмов с использованием циклов и ветвлений для управления формальными исполнителями, такими как Робот, Черепашка, Чертёжник.
- Выполнение алгоритмов вручную и на компьютере. Синтаксические и логические ошибки.
- Ветвления. Составные условия (запись логических выражений на изучаемом языке программирования). Цикл с условием. Цикл с переменной.
- Определение возможных результатов работы алгоритма при данном множестве входных данных, определение возможных входных данных, приводящих к данному результату.

В 8 классе ученики начинают изучать языки программирования. При освоении этой темы высокую эффективность продемонстрировало применение онлайн-ресурсов, цифровых приложений и сетевых сервисов.

Особое внимание требуется уделить заданиям на составление и программирование базовых алгоритмических структур на разных языках программирования.

При изучении курса информатики в 9 классе особое внимания требуют следующие темы:

Модель. Задачи, решаемые с помощью моделирования. Классификации моделей. Материальные (натурные) и информационные модели. Непрерывные и дискретные модели. Имитационные модели. Игровые модели. Оценка адекватности модели моделируемому объекту и целям моделирования.

Табличные модели. Таблица как представление отношения.

При изучении темы «Алгоритмы и программирование» рекомендуется применять метод «Самостоятельная разработка программ». Это помогает развить навыки самоконтроля и планирования, а также улучшает предметные и метапредметные умения, формировать умения анализировать задачи, разбивать их на подзадачи и планировать последовательность действий для их решения. Это помогает развивать логическое мышление и способность к анализу.

Учителям на уровне среднего общего образования:

С целью совершенствования преподавания учебного предмета «Информатика» в 10-11 классах в качестве рекомендаций предлагается:

- обращать внимание, прежде всего, на ключевые базовые темы, включённые в ЕГЭ;
- для успешного решения заданий необходимо глубокое понимание разделов информатики: «Логика и алгоритмы» и «Построение алгоритмов и практические вычисления», «Программирование»;
- при изучении содержательного раздела «Программирование» использовать возможности онлайн ресурсов, цифровых приложений, сетевых сервисов;
- использовать материалы банка заданий ЕГЭ, опубликованные в открытом сегменте ЕГЭ на сайте ФИПИ (<http://www.fipi.ru>);
- обратить внимание при организации внеурочной деятельности обучающихся на имеющиеся в Иркутской области организации дополнительного образования, ориентированные на развитие цифровых навыков: «Кванториум», «IT-кубы», «Точки роста».

При изучении курса информатики в 10 классе особое внимание требуется уделить следующим темам:

- Информация, данные и знания. Универсальность дискретного представления информации.
- Алгебра логики. Высказывания. Логические операции. Таблицы истинности логических операций «дизъюнкция», «конъюнкция», «инверсия», «импликация», «эквиваленция». Логические выражения. Вычисление логического значения составного высказывания при известных значениях входящих в него элементарных высказываний. Таблицы истинности логических выражений. Логические операции и операции над множествами.

При изучении курса информатики в 11 классе особое внимание требуется уделить следующим темам:

- Модели и моделирование. Цели моделирования. Соответствие модели моделируемому объекту или процессу. Формализация прикладных задач.
- Этапы решения задач на компьютере. Язык программирования (Паскаль, Python и др.). Основные конструкции языка программирования.
- Анализ данных.

○ *ИПК / ИРО, иным организациям, реализующим программы профессионального развития учителей*

Муниципальным методическим службам:

Рекомендуется спланировать методическую работу учителей информатики муниципальном уровне, включив в нее:

- Работу со статистико-аналитическими материалами по результатам ЕГЭ в 2025 года для выявления дефицитов школьников муниципалитета в освоении федеральной рабочей программы по информатике на всех уровнях преподавания. Методическим объединениям школьного/муниципального/регионального уровней детально изучить методический анализ результатов ЕГЭ в Иркутской области за 2025 год и информационно-методическое письмо об особенностях преподавания учебного предмета «Информатика» в 2025 – 2026 учебном году. Провести не менее 1 муниципального мероприятия.

- Муниципальное / межшкольное мероприятие по сопоставлению перечня (кодификатора) распределенных по классам проверяемых требований к результатам освоения основной образовательной программы НОО / ООО / СОО и элементов содержания по информатике с текущими результатами и результатами внешних оценочных процедур по информатике на муниципальном уровне, уделяя особое внимание работе с педагогами общеобразовательных учреждений с небольшим (по штатному расписанию) количеством учителей, наличием молодых специалистов, педагогов с непрофильным базовым образованием, с педагогами школ отдаленных, труднодоступных районов.

- Изучение ситуации с преподаванием информатики и организацией учебного процесса в ОО, показавших наиболее низкие результаты ЕГЭ по информатике.

- Серию очных методических мероприятий по устранению выявленных адресных дефицитов школьников с разным уровнем подготовки, направленных на освоение педагогами обоснованных форм, эффективных методов повышения качества обученности педагогами, обучающиеся которых показывают высокие образовательные результаты, в том числе открытые уроки, мастер-классы, методические школы и др. (не менее двух мероприятий).

- Обобщение опыта работы учителей информатики ОО, показавших наиболее высокие результаты ЕГЭ по информатике.

- Очные муниципальные мероприятия, направленные на повышение квалификации учителей информатики по формированию предметной и оценочной компетентности учителей информатики. Организовывать для учителей

информатики мероприятия разного уровня (августовский педагогический совет, педагогический форум, стратегическая сессия и т.п.), посвященные проблематике ЕГЭ и распространению педагогического опыта, обеспечивающие достижение высоких результатов по информатике, подтверждаемых на ГИА.

– Контроль выполнения составленного плана повышения качества обученности по информатике общеобразовательными организациями с фиксацией устранения/уменьшения образовательных дефицитов, выявленных у обучающихся при проведении контрольно-оценочных процедур.

ИРО, иным организациям, реализующим программы профессионального развития учителей:

Рекомендуется спланировать методическую работу учителей информатики на региональном уровне, включив в нее:

– Работу со статистико-аналитическими материалами по результатам ЕГЭ 2025 года для выявления типичных дефицитов школьников в освоении федеральной рабочей программы по информатике на всех уровнях преподавания, фиксации типичных ошибок учащихся.

– Цикл мероприятий для учителей по устранению учебных дефицитов школьников с разным уровнем подготовки в соответствии с перечнем (кодификатором) распределенных по классам проверяемых требований к результатам освоения основной образовательной программы НОО / ООО / СОО и элементов содержания по информатике с результатами внешних оценочных процедур, уделяя особое внимание работе с педагогами общеобразовательных учреждений с небольшим (по штатному расписанию) количеством учителей, наличием молодых специалистов, педагогов с непрофильным базовым образованием, педагогами школ отдаленных, труднодоступных районов.

– Повышение квалификации по дополнительным профессиональным программам в соответствии с выявленными в ходе анализа результатов ГИА по информатике типичными ошибками обучающихся. Организовать практические занятия для педагогов на базе общеобразовательных организаций региона по совершенствованию преподавания учебного предмета «Информатика».

- Цикл мероприятий для учителей по использованию эффективных форм, приемов работы, демонстрацию лучших практик обучения информатике, в том числе мастер-классы, практикумы, методические школы и др.
- Создание условий для обобщения и транслирования управленческого и педагогического опыта работы коллективов общеобразовательных организаций, демонстрирующих высокий уровень учебных достижений при наличии незначительных рисков, на другие общеобразовательные организации Иркутской области.
- Разработку и реализацию индивидуальных образовательных маршрутов для педагогов, испытывающих профессиональные затруднения, направленные на ликвидацию затруднений в зависимости от выявленного уровня.

2.1.2....по организации дифференцированного обучения школьников с разными уровнями предметной подготовки

○ *Учителям*

В целях обеспечения дифференцированной подготовки рекомендуется проводить диагностические работы (по завершении изучения тем и разделов). Такой промежуточный контроль призван диагностировать состояние знаний по изученному материалу, а также степень сформированности проверяемых умений.

В работе с обучающимися с уровнем подготовки ниже среднего необходима работа с базовыми информационными понятиями и конструкциями, возможно использование технологии уровневой дифференциации, в которой реализуется принцип коррекции знаний, что дает возможность обучающимся усваивать не только базовый минимум стандарта образования, но и продвигаться на более высокий уровень.

В работе с обучающимися с минимальным начальным уровнем подготовки необходима работа с текстом на уроках информатики для формирования умения анализировать прочитанный текст (условие задачи), сделать из него выводы и составить алгоритм выполнения данного задания, несколько раз проверить свое решение (самоконтроль и самоорганизация).

Группа обучающихся со средним уровнем подготовки нуждается в дополнительной работе с алгоритмическим и программируемым материалом, в выполнении большого количества различных заданий, предполагающих преобразование и интерпретацию информации. Приоритетной технологией здесь может стать совместное обучение – технология сотрудничества. Обратить внимание на базовые конструкции программ.

Для группы обучающихся с повышенным и высоким уровнями подготовки можно использовать технологию «перевернутого» обучения. В процессе обучения эти обучающиеся проявляют мотивацию к изучению информатики и, как правило, обладают достаточными знаниями для серьёзной самостоятельной работы. Данной группе необходимо серьёзная внеурочная, факультативная и т.п. работа под руководством специально подготовленных преподавателей. Необходимо постоянное поддержание интереса и мотивации; развитие мышления ученика, через решение задач нестандартных и повышенной сложности, головоломок, участие в олимпиадах. Обучающихся высокомотивированных включать в олимпиадное движение, включить в дистанционное обучение на платформе Образовательного центра «Сириус».

○ *Администрациям образовательных организаций*

– Обеспечивать необходимые материально-технические условия реализации требований ФГОС СОО и образовательных программ по информатике.

– Укреплять материально-техническую базу кабинетов информатики и проводить планомерную работу по созданию современной информационно-образовательной среды в общеобразовательной организации.

– Организовать обсуждение результатов ГИА 2025 года на педагогическом совете, методическом совете общеобразовательной организации.

– Скорректировать методическую работу с педагогами по разноуровневой подготовке обучающихся по информатике. Разработать стратегию взаимодействия учителей на уровне начального общего образования, основного общего образования и среднего общего образования с целью проведения системной работы по формированию и развитию метапредметных умений обучающихся на всех этапах обучения.

– Обеспечить непрерывную систему повышения квалификации учителей для профессионального роста.

○ *ИПК / ИРО, иным организациям, реализующим программы профессионального развития учителей*

– провести контекстный анализ ЕГЭ 2025 года в разрезе каждого муниципалитета и выявить факторы риска учебной неуспешности.

- В рамках индивидуального образовательного маршрута педагога обеспечить повышение квалификации по ликвидации имеющихся профессиональных затруднений с использованием различных форм, таких как очные и дистанционные курсы повышения квалификации, педагогические мастерские, семинары, мастер-классы и др.
- Обобщить опыт работы учителей информатики на региональном уровне по организации дифференцированного обучения обучающихся с разными уровнями предметной подготовки по информатике.
- Организовать трансляцию эффективных педагогических практик образовательных организаций с наиболее высокими результатами ЕГЭ 2025 года. Проведение методических объединений в муниципальных единицах с обсуждением результатов ЕГЭ-2025 и выявлением причин неуспешности участников экзамена при выполнении того или иного задания.
- Способствовать распространению в педагогическую практику тех методик и технологий обучения, которые подтвердили свою эффективность.

2.2. Рекомендуемые темы для обсуждения / обмена опытом на методических объединениях учителей-предметников, в том числе по трансляции эффективных педагогических практик ОО с наиболее высокими результатами

- Анализ результатов ЕГЭ 2025 года в разрезе общеобразовательной организации с выявлением «проблемных полей» в знаниях выпускников для последующей методической корректировки процесса преподавания информатики;
- Работа простейших алгоритмов управления исполнителями и вычислительных алгоритмов;
- Создание программы (10–20 строк) для обработки символьной информации;
- Обработка целочисленной информации с использованием сортировки;
- Создание программы (20–40 строк) для анализа числовых последовательностей.

2.3. Рекомендуемые направления повышения квалификации работников образования

При организации мероприятий, направленных на повышение квалификации и профессионального мастерства учителей информатики рекомендованы следующие направления:

– повышение оценочной компетентности (технологии проверки заданий с развернутым ответом с целью ознакомления учителей-предметников с процедурой оценивания письменной и устной части внешних оценочных процедур; особенности проверки заданий разных типов; применение в работе технологии формирующего оценивания, критериального оценивания).

- организация работы с учащимися с рисками учебной неуспешности;
- организация работы с одаренными детьми в области информатики;
- использование эффективных методик, технологий, приёмов, цифровых ресурсов и др. обучения информатике.

СОСТАВИТЕЛИ АНАЛИЗА по учебному предмету:

Специалисты, привлекаемые к анализу результатов ЕГЭ по учебному предмету:

Лебедева Светлана Юрьевна

Специалисты, привлекаемые к подготовке методических рекомендаций на основе результатов ЕГЭ по учебному предмету:

Перминова Оксана Викторовна, Гусева Алёна Владимировна