

АНАЛИЗ РЕЗУЛЬТАТОВ ВЫПОЛНЕНИЯ ЗАДАНИЙ КИМ ЕГЭ ПО ИНФОРМАТИКЕ

1.1. Анализ выполнения заданий КИМ

1.1.1. Статистический анализ выполнения заданий КИМ в 2025 году

1.1.1.1. Основные статистические характеристики выполнения заданий КИМ в 2025 году

Основные статистические характеристики выполнения заданий в целом представлены в Таб.1. 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.

Таблица 1

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Республике Саха (Якутия) 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	84,4	63,8	90,6	95,6	98,6
	2024 г.		83,8	61,6	89,6	97,2	97,4
2	Умение строить таблицы истинности и логические схемы	Б	65,8 ↓	26,8 ↓	73,8 ↓	93,0	95,9
	2024 г.		76,5	38,4	87,5	97,6	97,4
3	Умение поиска информации в реляционных базах данных	Б	69,5 ↑	43,0 ↑	74,2 ↑	87,5 ↑	97,3 ↑
	2024 г.		58,8	30,8	65,3	75,2	92,1
4	Умение кодировать и декодировать информацию	Б	81,1	54,7 ↓	89,8	94,1	98,6
	2024 г.		85,8	65,7	91,7	96,9	97,4

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Республике Саха (Якутия) в группах участников экзамена с разными уровнями подготовки				
			средний, %	в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
5	Формальное исполнение простого алгоритма, записанного на естественном языке, или умение создавать линейный алгоритм для формального исполнителя с ограниченным набором команд, или умение восстанавливать исходные данные линейного алгоритма по результатам его работы	Б	28,8 ↓	1,7 ↓	14,1 ↓↓↓	73,2 ↓	97,3
	2024 г.		45,6	9,2	44,8	84,6	97,4
6	Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов	Б	40,3 ↑	12,0 ↑	43,2 ↑↑	59,9 ↑	83,6 ↓
	2024 г.		25,3	5,1	21,3	49,2	89,5

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Республике Саха (Якутия) 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.
7	Умение определять объём памяти, необходимый для хранения графической и звуковой информации	Б	51,2 ↑↑	8,8 ↑	53,3 ↑↑	89,7 ↑↑	97,3
	2024 г.		30,9	2,9	26,4	65,4	94,7
8	Знание основных понятий и методов, используемых при измерении количества информации	Б	34,4 ↑	2,8	25,0 ↑	77,6 ↑↑	90,4
	2024 г.		21,8	2,9	12,5	54,7	86,8
9	Умение обрабатывать числовую информацию в электронных таблицах	Б	19,8	1,1	10,7	44,1 ↓	82,2 ↓
	2024 г.		23,4	1,3	13,9	60,2	92,1
10	Информационный поиск средствами текстового процессора	Б	69,8 ↑	43,9 ↑	75,0 ↑	86,0 ↑	97,3 ↑
	2024 г.		53,4	28,3	55,4	74,8	92,1
11	Умение подсчитывать информационный объём сообщения	П	19,9 ↑	4,0	14,1 ↑	36,0 ↑	76,7 ↑↑↑
	2024 г.		10,5	1,3	6,6	26	36,8

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Республике Саха (Якутия) 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.
12	Умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд	П	28,3 ↓↓	1,1 ↓	18,8 ↓↓↓↓	65,4 ↓↓	87,7 ↓
	2024 г.		56,3	13,3	60,5	94,5	97,4
13	Умение использовать маску подсети	П	29,3 ↑	3,1	18,6	65,8 ↑	94,5 ↑
	2024 г.		22,8	0,3	14,2	59,1	86,8
14	Знание позиционных систем счисления	П	30,4 ↑	3,1	17,0 ↑	74,3 ↑	91,8 ↓
	2024 г.		18,6	0,6	5,1	57,1	97,4
15	Знание основных понятий и законов математической логики	П	35,5 ↑	3,1	23,4 ↑	83,1 ↑↑	98,6
	2024 г.		22,8	1,3	12,3	60,6	94,7
16	Вычисление рекуррентных выражений	П	39,5 ↓	7,7	33,8 ↓	76,8 ↓	93,2 ↓
	2024 г.		45,8	6	42,9	93,3	97,4
17	Умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования	П	13,7 ↓	0,0	1,8	34,6 ↓↓	86,3 ↓
	2024 г.		18	0	4,4	56,7	97,4

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Республике Саха (Якутия) 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.
18	Умение использовать электронные таблицы для обработки целочисленных данных	П	29,6 ↓	2,3	22,3 ↓	61,8 ↓	93,2
	2024 г.		34,7	4,4	30	72,8	94,7
19	Умение анализировать алгоритм логической игры	Б	61,3 ↓	26,5	63,9 ↓	91,2	100,0 ↑
	2024 г.		66,8	29,2	75	92,1	94,7
20	Умение найти выигрышную стратегию игры	П	37,8 ↓	3,1 ↓	30,1 ↓↓↓	82,0 ↓	94,5
	2024 г.		54,4	9,8	60,2	91,3	97,4
21	Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию	В	36,1 ↓	3,1	27,3 ↓	79,0	95,9 ↑
	2024 г.		40,3	3,8	38,7	81,5	89,5
22	Построение математических моделей для решения практических задач. Архитектура современных компьютеров. Многопроцессорные системы	П	25,3 ↑	4,3	21,9 ↑	44,5 ↑	79,5 ↑↑↑

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Республике Саха (Якутия) 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.
	2024 г.		10,8	2,2	6,1	25,6	47,4
23	Умение анализировать ход исполнения алгоритма	П	33,9 ↓	1,7	23,2 ↓	80,1 ↓	91,8 ↓
	2024 г.		37,1	4,1	30	83,9	97,4
24	Умение создавать собственные программы (10–20 строк) для обработки символьной информации	В	2,2	0,0	0,2	1,5	30,1 ↑
	2024 г.		0,8	0	0,2	0,8	15,8
25	Умение создавать собственные программы (10–20 строк) для обработки целочисленной информации	В	4,1 ↓	0,0	0,2	3,7 ↓ ↓	52,1 ↓ ↓
	2024 г.		10,4	0	1,5	31,5	78,9
26	Умение обрабатывать целочисленную информацию с использованием сортировки	В	3,1	0,0	0,4	2,9 ↓	37,0 ↓
	2024 г.		3,7	0	0,4	7,9	53,9

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Республике Саха (Якутия) в группах участников экзамена с разными уровнями подготовки				
			средний, %	в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
27	Умение выполнять последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и построение модели, преобразование данных, визуализация данных, интерпретация результатов	В	6,7	0,0	0,6	10,5	67,1
	2024 г.		-	-	-	-	-
Используемые в таблице обозначения: Закрашенная ячейка – выявленные сложные для участников задания; Базовый уровень – до 50% выполнения участниками экзамена; Повышенный уровень – до 15% выполнения участниками экзамена; Высокий уровень – до 15% выполнения участниками экзамена. ↓ – понижение значения показателя, по сравнению с предыдущим годом более 5 % на задачи базового уровня и более 3 % на задачи повышенной и высокого уровня сложности, красным отмечены понижения на более 10 %; ↑ – повышение значения показателя, по сравнению с предыдущим годом более 5 % на задачи базового уровня и более 3 % на задачи повышенной и высокого уровня сложности, зеленым отмечены повышения на более 10 %.							

Таблица 2

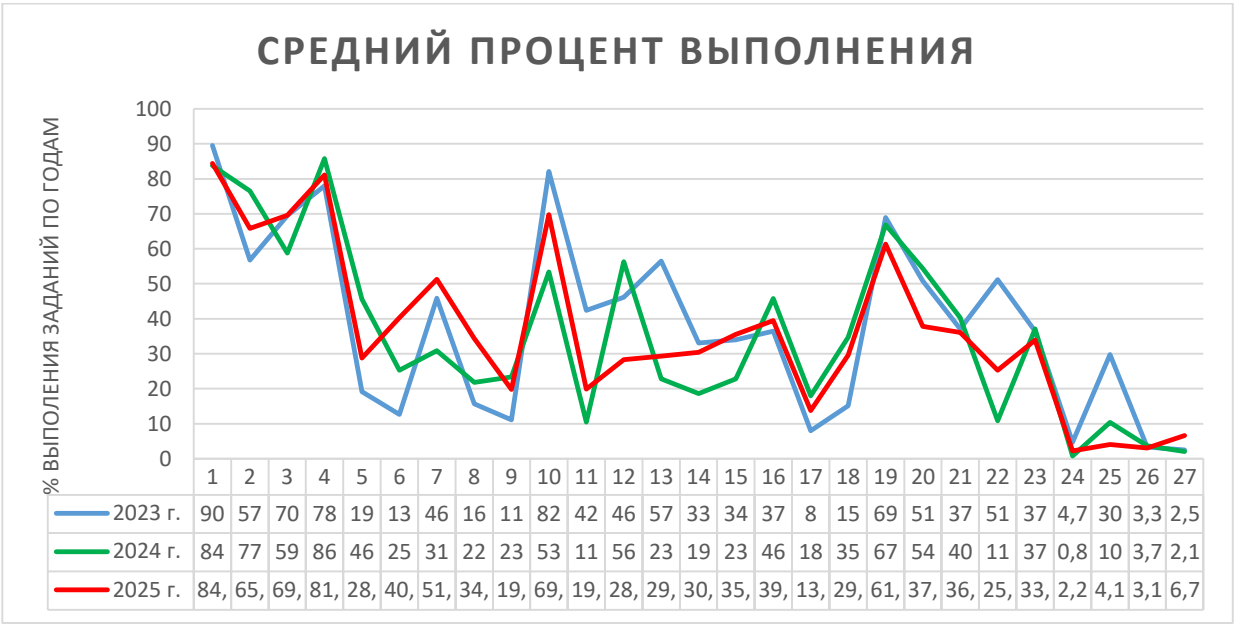
Номер задания / критерия оценивания в КИМ	Количество полученных первичных баллов	Процент участников экзамена в Республике Саха (Якутия), получивших соответствующий первичный балл за выполнения задания в группах участников экзамена с разными уровнями подготовки			
		в группе не преодолевших минимальный 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	0	36,2	9,4	4,4	1,4
	1	63,8	90,6	95,6	98,6
2	0	73,2	26,2	7,0	4,1
	1	26,8	73,8	93,0	95,9
3	0	57,0	25,8	12,5	2,7
	1	43,0	74,2	87,5	97,3
4	0	45,3	10,2	5,9	1,4
	1	54,7	89,8	94,1	98,6
5	0	98,3	85,9	26,8	2,7
	1	1,7	14,1	73,2	97,3
6	0	88,0	56,8	40,1	16,4
	1	12,0	43,2	59,9	83,6
7	0	91,2	46,7	10,3	2,7
	1	8,8	53,3	89,7	97,3
8	0	97,2	75,0	22,4	9,6
	1	2,8	25,0	77,6	90,4
9	0	98,9	89,3	55,9	17,8
	1	1,1	10,7	44,1	82,2
10	0	56,1	25,0	14,0	2,7
	1	43,9	75,0	86,0	97,3
11	0	96,0	85,9	64,0	23,3
	1	4,0	14,1	36,0	76,7
12	0	98,9	81,3	34,6	12,3
	1	1,1	18,8	65,4	87,7
13	0	96,9	81,4	34,2	5,5
	1	3,1	18,6	65,8	94,5
14	0	96,9	83,0	25,7	8,2
	1	3,1	17,0	74,3	91,8
15	0	96,9	76,6	16,9	1,4
	1	3,1	23,4	83,1	98,6
16	0	92,3	66,2	23,2	6,8

Номер задания / критерия оценивания в КИМ	Количество полученных первичных баллов	Процент участников экзамена в Республике Саха (Якутия), 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	7,7	33,8	76,8	93,2
17	0	100,0	98,2	65,4	13,7
	1	0,0	1,8	34,6	86,3
18	0	97,7	77,7	38,2	6,8
	1	2,3	22,3	61,8	93,2
19	0	73,5	36,1	8,8	0,0
	1	26,5	63,9	91,2	100,0
20	0	96,9	69,9	18,0	5,5
	1	3,1	30,1	82,0	94,5
21	0	96,9	72,7	21,0	4,1
	1	3,1	27,3	79,0	95,9
22	0	95,7	78,1	55,5	20,5
	1	4,3	21,9	44,5	79,5
23	0	98,3	76,8	19,9	8,2
	1	1,7	23,2	80,1	91,8
24	0	100,0	99,8	98,5	69,9
	1	0,0	0,2	1,5	30,1
25	0	100,0	99,8	96,3	47,9
	1	0,0	0,2	3,7	52,1
26	0	100,0	99,2	95,2	52,1
	1	0,0	0,8	3,7	21,9
	2	0,0	0,0	1,1	26,0
27	0	100,0	99,0	83,8	21,9
	1	0,0	0,8	11,4	21,9
	2	0,0	0,2	4,8	56,2

На диаграмме 1 представлен средний процент выполнения заданий ЕГЭ по информатике в 2023-2025 годах.

Диаграмма 1

Процент участников, справившихся с заданиями ЕГЭ по информатике 2023-2025 годах.



На диаграмме 2 представлена разность средних процентов выполнения заданий ЕГЭ по информатике.

Диаграмма 2

Изменение процента участников, справившихся с заданиями ЕГЭ по информатике.

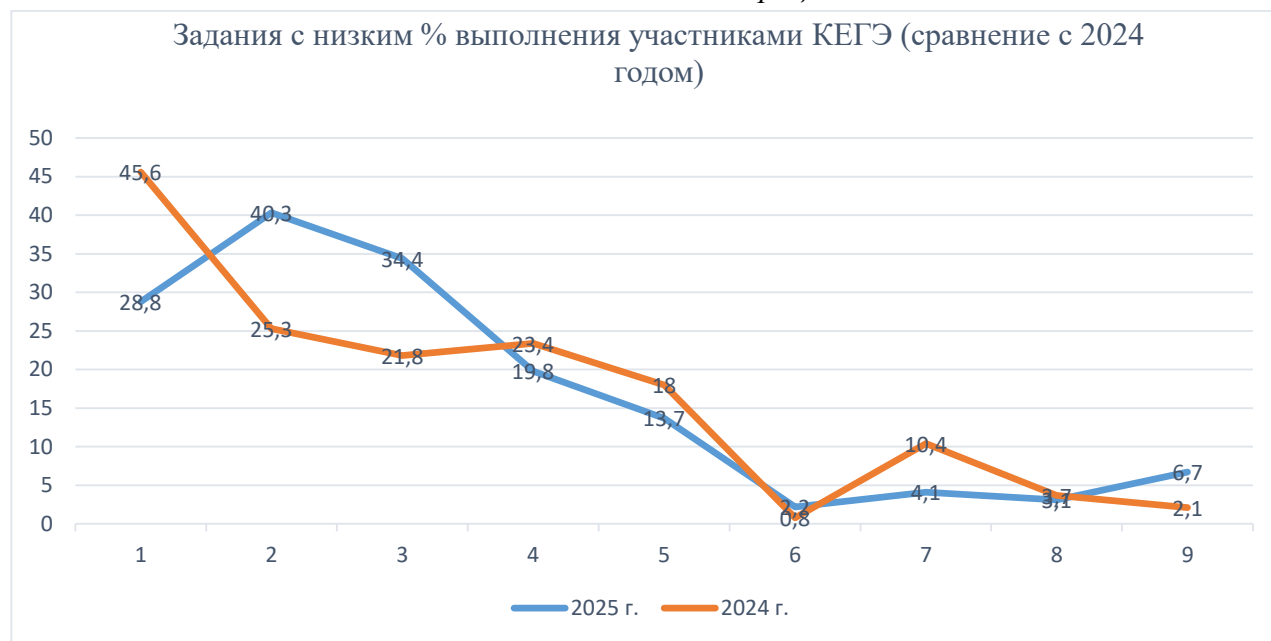


1.1.1.2. Выявление сложных для участников ЕГЭ заданий

Наиболее сложными заданиями для участников КЕГЭ по информатике в 2025м году (ниже 50% для базового, ниже 15% для повышенного и высокого уровней сложности) оказались задания № 5, 6, 8, 9, 17, 24 - 27.

На диаграмме 3 представлен процент выполнения сложных заданий участниками ЕГЭ 2024 и 2025 годах.

Диаграмма 3.
Проценты выполнения сложных заданий.



На таблице 3 показана сравнительная динамика выполнения заданий с низким процентом выполнения за 2024-2025 гг.

Таблица 3
Сравнительная динамика выполнения заданий с низким процентом выполнения за 2024-2025 гг.

№ задания	5	6	8	9	17	24	25	26	27
Уровень сложности	Б	Б	Б	Б	П	В	В	В	В
2025 г.	28,8	40,3	34,4	19,8	13,7	2,2	4,1	3,1	6,7
2024 г.	45,6	25,3	21,8	23,4	18	0,8	10,4	3,7	2,1
Изменения	-16,8	15	12,6	-3,6	-4,3	1,4	-6,3	-0,6	4,6

В рамках выполнения анализа, укажем линии заданий с наименьшими процентами выполнения.

○ Задания базового уровня (с процентом выполнения ниже 50):

1) **Задание № 5** (базовый уровень) согласно кодификатору и спецификации КИМ КЕГЭ 2025 года проверяет 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. Средний процент выполнения - 28,8%, по сравнению с 2024 годом процент уменьшился на 16,8%. Понижение обусловлено тем, что в задании рассматривались нетрадиционные системы счисления, например, троичная, пятеричная система счисления.

2) **Задание № 6** (базовый уровень) проверяет определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Средний процент выполнения - 40,3% (25,3% - в 2024 году, повышение на 15%) - пошаговое построение фигуры в Кумире или на листе, нарисовать и посчитать точки. По сравнению с прошлым годом процент выполнения значительно повысился.

3) **Задание № 8** (базовый уровень) определяет знание основных понятий и методов, используемых при измерении количества информации. Процент выполнения 34,4% (21,8% - в 2024 году), по сравнению с прошлым годом повысился на 12,6%. Повышение обусловлено тем, что обучающиеся научились применять навыки программирования при решении данного задания.

4) **Задание № 9** (базовый уровень) направлено на проверку умения обрабатывать числовую информацию в электронных таблицах, выполняется с использованием прилагаемых файлов. Процент выполнения - 19,8% (23,4% - в 2024 году), уменьшение на 3,6%. Задание сильно не изменилось с прошлого года.

В заданиях базового уровня во всех группах затруднения вызвали задания № 5, № 6 (на алгоритмы), № 8 (на измерение количества информации), № 9 (на обработку табличной информации). При этом по сравнению с прошлым годом во всех группах наблюдается повышение процента выполнения в заданиях № 6, № 8 и понижение в заданиях № 5, № 9. Эти задания комплексно проверяют предметную и метапредметную подготовку выпускников к экзамену, вполне доступны для решения всеми группами участников КЕГЭ.

○ **Задания повышенного и высокого уровня (с процентом выполнения ниже 15%):**

1) **Задание № 17** (повышенный уровень) проверяет умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования. Процент выполнения - 13,7%, снижение на 4,3%.

2) **Задание № 24** (высокого уровня) проверяет умение создавать собственные программы (10–20 строк) для обработки символьной информации. Процент выполнения - 2,2% (0,8% - в 2024 году, повышение на 1,4%), условие задания немного изменилось, при его решении есть большое преимущество у языка программирования Python.

3) **Задание № 25** (высокого уровня) проверяет умение создавать собственные программы (10–20 строк) для обработки целочисленной информации. Процент выполнения - 4,1% (10,4% - в 2024 году, понижение на 6,3%), проценты выполнения понижается в последние три года.

4) **Задание № 26** (высокого уровня) проверяет умение обрабатывать целочисленную информацию с использованием сортировки. Процент выполнения - 3,1% (3,7% - в 2024 году, понижение на 0,6%), задание сложное, олимпиадного уровня, требует использования компаратора для сортировки и специальных конструкций структур или пар.

5) **Задание № 27** (высокого уровня) проверяет 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. Задание новое. Процент выполнения - 6,7%.

В заданиях повышенного уровня затруднение вызвало одно задание № 17, где по сравнению с прошлым годом понизился процент выполнения во всех рассматриваемых группах. Задания № 17 из раздела «Алгоритмы и программирование», связано с математикой, с метапредметной способностью к алгоритмическому мышлению. Не все ребята смогли выполнить данное задание. Процент выполнения составил 13,7%. В группе не преодолевших порог задание никто не выполнил, в группе от минимального до 60 баллов. выполнили только 1,8%, в группе от 61 до 80 баллов - ухудшение на 22,1%, а для группы от 81 до 100 баллов - ухудшение на 11,1% по сравнению с прошлым годом.

Самыми затруднительными являются задания № 24, № 25, № 26, № 27 – задания высокого уровня сложности на темы «Программирование» и «Информационные технологии», в которых требуется разработка алгоритма и написание программы для обработки символьной информации (задание № 24), числовой информации (задание № 25), обработки числовой информации с использованием сортировки (задание № 26), и выполнение последовательности решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и построение модели, преобразование данных, визуализация данных, интерпретация результатов (задание № 27). Отдельные задания из этой группы можно решать с помощью электронных таблиц, причем конкретный способ решения выбирает участник. Для решения всех перечисленных заданий участник должен иметь опыт программирования и разработки алгоритмов. Слабые участники значительно справились хуже (или совсем не справились) с этими заданиями по сравнению с учениками, показавшими высокие тестовые баллы. Ни один участник из группы, не преодолевшей минимальный порог, не выполнил ни одного из этих заданий. С заданием № 24 выпускники в группе от 81 до 100 баллов справились лучше, чем в предыдущем году. В каждой группе наблюдается ухудшение процента выполнения задания по заданиям № 25 и № 26, по сравнению с прошлым годом.

В таблице 4 показано изменение процента выполнения заданий по группам, отрицательный знак означает ухудшение, положительный – улучшение показателя по сравнению с прошлым годом.

Таблица 4
Изменение процента выполнения заданий.

№ задания	Изменение среднего процента	в группе не преодолевших минимальный балл	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
5	-16,8	-7,5	-30,7	-11,4	-0,1
6	15,0	6,9	21,9	10,7	-5,9
8	12,6	-0,1	12,5	22,9	3,6
9	-3,6	-0,2	-3,2	-16,1	-9,9
17	-4,3	0,0	-2,6	-22,1	-11,1
24	1,4	0,0	0,0	0,7	14,3
25	-6,3	0,0	-1,3	-27,8	-26,8
26	-0,6	0,0	0,0	-5,0	-16,9

○ Прочие задания

Отметим ухудшение процента выполнения

- задания № 2 базового уровня (умение строить таблицы истинности и логические схемы) на 10,7%; для группы, не преодолевшей минимальный балл, - на 11,6 %, от минимального до 60 баллов - на 13,7%;

- задания № 5 базового уровня (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) на 16,8% для группы, не преодолевшей минимальный балл, - на 7,5%, от минимального до 60 баллов - на 30,7%, от 61 до 80 баллов - на 11,4%;

- задания № 12 повышенного уровня (умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд) на 28,0%, для группы, не преодолевшей минимальный балл, - на 12,2%, от минимального до 60 баллов - на 41,8%, от 61 до 80 баллов - на 29,1%, для группы от 81 до 100 баллов - на 9,7%;

- задания № 20 повышенного уровня (умение найти выигрышную стратегию игры) на 16,6%, для группы, не преодолевшей минимальный балл, - на 6,7%, от минимального до 60 баллов - на 30,1%, от 61 до 80 баллов - на 9,3%, для группы от 81 до 100 баллов - на 2,9%.

Сложившаяся ситуация объясняется непривычной формулировкой, объёмным текстом и недостаточным вниманием при выполнении математических подсчётов.

1.1.1.3. Прочие результаты статистического анализа

Наиболее успешно усвоены задания базового уровня: № 1 (84,4%), № 3 (69,5%, увеличение на 10,2), № 10 (69,8%, увеличение на 16,4%). Повышенного уровня: № 14 (30,4%, увеличение на 11,8%), № 15 (35,5%, увеличение на 12,7%). Здесь наблюдается положительная динамика выполнения данных заданий по сравнению с прошлым годом во всех рассматриваемых группах участников.

Отметим задание базовой сложности № 7: улучшение процента выполнения на 20,3% (с 30,9% до 51,2%) для группы от минимального до 60 баллов - на 26,9%, от 61 до 80 баллов - на 24,3%, от 81 до 100 баллов - на 2,6%; а также улучшение процента выполняемости наблюдается у задания повышенной сложности № 22 улучшение на 14,5% (с 10,8% до 25,3%) для группы от минимального до 60 баллов - на 15,8%, от 61 до 80 баллов - на 18,9%, от 81 до 100 баллов - на 32,1%.

Для группы от 81 до 100 т.б. заметим повышение процента выполнения на 39,9% задания базового уровня сложности № 11 и на 14,3% задания высокого уровня сложности № 24.

1.1.2. Содержательный анализ выполнения заданий КИМ

Статистический анализ выполнения заданий КИМ и результатов экзамена по информатике позволил выделить ряд заданий, вызвавших наибольшие сложности у участников экзамена.

Приведен содержательный разбор заданий, вызвавших наибольшие сложности у участников экзамена, на основе материалов открытого в регионе **варианта 325**.

Задание № 5

На 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. Задание **базового** уровня.

Содержание задания:

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 .

Результат переводится в десятичную систему и выводится на экран.

Например, для исходного числа $1110 = 1023$ результатом является число $1021013 = 30710$, а для исходного числа $610 = 203$ это число $20203 = 6010$.

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

Решение: Задание можно решить аналитически, поскольку число R получается дописыванием 2-значного или 3-хзначного числа, то минимальное число N будет делиться на 3 с остатком 2. Значит ищем N так, чтобы было R был ближайшим 6-значным в троичной системе счисления, $R = abc101 \geq 22222$, N должно быть больше 9 и иметь остаток 2 при делении на 3, значит $N = 11$.

Также можно написать программу. Решим задачу перебором. Перебираем N . Для каждого N найдем R . Как только R станет больше 242, прекращаем цикл для перебора N и ответом будет тот N , при котором мы прекратили цикл. Таким образом, если N кратно 3, то $R = N * 9 + N \% 9$; в противном случае, если остаток от деления на 3 равен 1, то $R = R * 9 + 5$; и если остаток от деления на 3 равен 2, то $R = R * 27 + 10$;

Программа на C++ выглядит следующим образом:

```
#include <iostream>
using namespace std;
int main()
{
    int R;
    for (int N = 1; N <= 1000; N++) {
        if (N % 3 == 0) R = N * 9 + N % 9;
        else if (N % 3 == 1) R = N * 9 + 5;
        else R = N * 27 + 10;
        if (R >= 242) { cout << N; break; }
    }
}
```

Ответ: 11.

Типичные ошибки при выполнении:

- Арифметические ошибки при переводе с одной системы счисления в другую.
- Невнимательное прочтение условия заданий: могут найти вместо максимума минимум и наоборот.
- Решение заданий по шаблону, без учета изменений в формулировках условий и самих вопросов.

Анализ возможных причин получения выявленных типичных ответов: Одной из существенных причин является неумение работать с длинными текстами, содержащими несколько условий. Также важно отметить, что это задание также достаточно легко решается с помощью программирования, при том, что решение методом рассуждений приводит к большему количеству ошибок из-за необходимости одновременного учета нескольких условий.

Пути устранения типичных ошибочных ответов в ходе обучения школьников: Педагогам и выпускникам необходимо разбирать несколько вариантов решения заданий разными способами, анализируя их особенности. Некоторые типы заданий решаются аналитически, а некоторые - с помощью программирования.

Задание № 6

На определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Задание **базового** уровня.

Содержание задания:

Исполнитель «Черепашка» действует на плоскости с декартовой системой координат. В начальный момент Черепашка находится в начале координат, её голова направлена вдоль положительного направления оси ординат, хвост опущен. При опущенном хвосте Черепашка оставляет на поле след в виде линии. В каждый конкретный момент известны 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 Черепашки на n единиц в том направлении, куда указывает её голова;

Назад n (где n – целое число), вызывающая передвижение в противоположном голове направлении;

Направо m (где m – целое число), вызывающая изменение направления движения на m градусов по часовой стрелке,

Налево m (где m – целое число), вызывающая изменение направления движения на m градусов против часовой стрелки.

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

Черепашке был дан для исполнения следующий алгоритм.

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

Поднять хвост

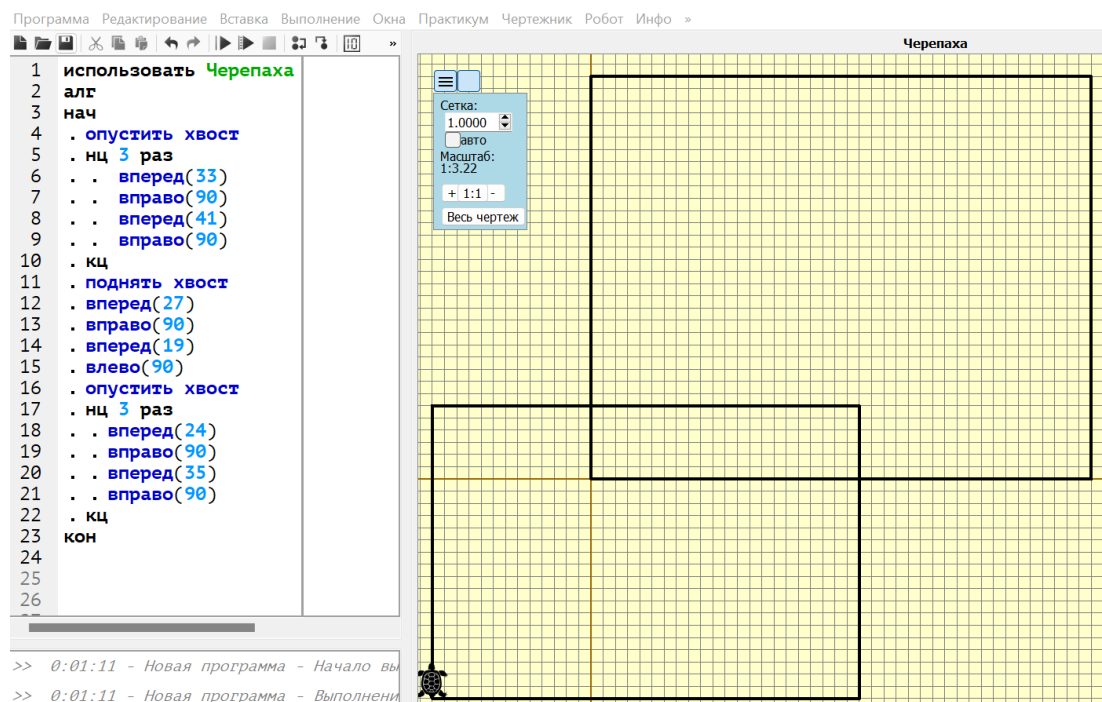
Вперёд 27 Направо 90 Вперёд 19 Налево 90

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.

Решение: Нарисуем фигуры на Кумире:



Получаем объединение прямоугольников с учетом пересечения. Количество точек равно: $34 \cdot 42 + 25 \cdot 36 - 7 \cdot 23 = 1428 + 900 - 161 = 2167$. Нужно учитывать границы, поэтому длинам сторон добавляем 1.

Ответ: 2167.

Типичные ошибки при выполнении:

- Вместо объединения находят пересечение фигур.
- Не учитывают границы фигур.
- Невнимательное прочтение условия заданий.

Анализ возможных причин получения выявленных типичных ответов: Причиной ошибки является неумение работать с длинными текстами, ошибки по невнимательности, отсутствие понимания объединения и пересечения множеств. Задание можно выполнить аналитически, но в этом нужно учитывать границы областей.

Пути устранения типичных ошибочных ответов в ходе обучения школьников: Разбор заданий, связанных с понятием множеств и операциями с ними. Решение заданий разными способами и анализ преимуществ каждого из них.

Задание № 8

На знание о методах измерения количества информации. Задание **базового** уровня.

Содержание задания:

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

1. ЕЕЕЕЕЕ
2. ЕЕЕЕЕИ
3. ЕЕЕЕЕО
4. ЕЕЕЕЕР
5. ЕЕЕЕЕТ
6. ЕЕЕЕЕЯ

.....

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.

Решение:

Задание решается переборным способом. Приведем решение на языке C++:

Переменными i, j, k, l, m, n обозначены соответствующие цифры шестеричной системы счисления, n1 это номер текущего числа, n2 - количество цифр 5 = 'Я', ans – ответ, самое максимальное число, удовлетворяющее условиям задания.

```
#include <iostream>
using namespace std;
int main()
{
    int i, j, k, l, m, n, n1 = 0, n2, ans;
    for (i = 0; i <= 5; i++)
        for (j = 0; j <= 5; j++)
            for (k = 0; k <= 5; k++)
                for (l = 0; l <= 5; l++)
                    for (m = 0; m <= 5; m++)
                        for (n = 0; n <= 5; n++)
                        {
                            n1++;
                            n2 = 0;
                            if (i == 5) n2++;
                            if (j == 5) n2++;
                            if (k == 5) n2++;
                            if (l == 5) n2++;
                            if (m == 5) n2++;
                            if (n == 5) n2++;
                            if ((i == 3 or i == 4 or i == 5) and n2 ==
1 and n1 % 2 == 0) ans = n1;
                        }

    cout << ans;
    return 0;
}
```

Ответ: 45100.

Типичные ошибки при выполнении:

- Арифметические ошибки.
- Невнимательное прочтение условия заданий.
- непонимание условия.
- Число не может начинаться с цифры 0.
- Не учитывают то, что нумерация списка начинается с 1, а не с 0.
- Решение заданий по шаблону, без учета изменений в формулировках условий и самих вопросов.

Анализ возможных причин получения выявленных типичных ответов: Для правильного решения, необходимо иметь хороший уровень знаний по математике. При этом необходимо учесть, что первая цифра не может быть 0, для корректности числа. Также причинами неверного выполнения этого задания может быть неправильно понятое условие задачи.

Пути устранения типичных ошибочных ответов в ходе обучения школьников: Использовать на уроках информатики различные математические задания на комбинаторику и

Ищем среднее значение неповторяющихся чисел строки:

Вычисляем строки, удовлетворяющие второму условию:

Применим фильтр для двух условий и получим все строки, осталось найти среднее арифметическое всех строк.

Среднее: 58,96413502

Выделяя все строки, получим: целая часть равна 58.

Ответ: 58.

Типичные ошибки при выполнении:

- Неверный выбор функций и формул в электронных таблицах, вследствие чего получаются вычислительные ошибки.
- Неверное использование логических операций.
- Невнимательное прочтение условия заданий.
- Решение заданий по шаблону, без учета изменений в формулировках условий и самих вопросов.

Анализ возможных причин получения выявленных типичных ответов: Недостаточный уровень знаний в применении табличных технологий, незнание приемов работы с функциями и ошибки при их использовании. Для успешного выполнения этого задания нужно уметь формулировать сложные логические условия, содержащие логические операции «И» и «ИЛИ», одновременно используя редактор электронных таблиц, а также знать элементарные сведения из школьного курса математики. Выпускники, не справившиеся с заданием, не смогли сформулировать логическое условие.

Пути устранения типичных ошибочных ответов в ходе обучения школьников: Использование на уроках и в индивидуальной работе с учениками практических заданий на электронные таблицы с применением разнообразных формул и функций, развитие логики.

Задание № 17

На умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования. Задание **повышенного** уровня сложности.

Содержание задания:

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. Под тройкой подразумевается три идущих подряд элемента последовательности.

Решение:

Пример программы на C++:

```
#define _CRT_SECURE_NO_WARNINGS //17
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;

int main() {
    freopen("325_17.txt", "r", stdin);
    long long x, y, z, x1, y1, z1, k = 0, m = -100001, m1 = -300003;
    vector<int> a;
    while (cin >> x)
    {
        a.push_back(x);
        if (m < x and abs(x) % 100 == 17) m = x;
    }
    for (int i = 0; i < a.size() - 2; i++)
    {
```

```

        x = a[i]; if (99 < abs(x) and abs(x) < 1000) x1 = 1; else x1 =
0;
        y = a[i + 1]; if (99 < abs(y) and abs(y) < 1000) y1 = 1; else y1
= 0;
        z = a[i + 2]; if (99 < abs(z) and abs(z) < 1000) z1 = 1; else z1
= 0;
        if (x + y + z < m and x1 + y1 + z1 == 1)
        {
            k++;
            if (x + y + z > m1) m1 = x + y + z;
        }
    }
    cout << k << ' ' << m1;
    return 0;
}

```

Ответ: 2781 85899.

Анализ возможных причин получения выявленных типичных ответов:

- Неумение работать с файлом и с большим количеством данных.
- Невнимательное прочтение условия задания.
- Не учитываются отрицательные элементы.
- Неумение создавать программы для реализации нахождения максимума/минимума.
- непонимание условия задания и формализации требуемых условий.

Пути устранения типичных ошибочных ответов в ходе обучения школьников:

Для решения таких заданий нужен опыт, навык считывания с файла, требуется хорошая математическая подготовка, развитое логическое мышление в сочетании с достаточными навыками программирования. Основная ошибка в данном задании вызвана тем, что не учтены все условия нужной последовательности, то есть ошибка в алгоритме решения задачи. Также ошибки могут быть при написании программы в связи с недостаточным уровнем знаний и практики использования конкретного языка программирования.

Задание № 24

Навык создания собственных программ (10–20 строк) для обработки символьной информации. Задание **высокого** уровня сложности.

Содержание задания:

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.

Решение:

Пример программы на C++:

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
using namespace std;
int main() {

    string s;
    freopen("325_24.txt", "r", stdin);

```

```

cin >> s;
vector<int> p;
for (int i = 0; i < s.size(); i++)
    if (s[i] == 'N') p.push_back(i);

int m1 = 0;
for (int i = 0; i < p.size() - 1; i++)
{
    int m = 0; int k = 0;
    for (int j = p[i]; j <= p[i + 1]; j++) {
        if (s[j] >= '0' and s[j] <= '9') k++;
        if (k == 55 and j == p[i + 1]) {m = p[i + 1] - p[i]; m1 =
max(m1, m);}
        else if (k == 56) {m = j - p[i]; m1 = max(m1, m); break;}
    }
}
cout << m1;
return 0;
}

```

Ответ: 132.

Типичные ошибки при выполнении:

- Неумение работать со строками.
- Незнание инструментов и языков программирования для работы с символами.
- Неумение создавать программы для реализации сложных математических алгоритмов.
- Невнимательное прочтение условия задания.
- Пренебрежение граничными условиями.
- непонимание условия задания и формализации требуемых условий.

Анализ возможных причин получения выявленных типичных ответов: Для решения заданий высокого уровня сложности требуется хорошая математическая подготовка, развитое логическое мышление в сочетании с достаточными навыками программирования. Основная ошибка в данном задании вызвана тем, что не учтены все условия нужной последовательности, т.е. ошибка в алгоритме решения задачи. Также ошибки могут быть при написании программы в связи с недостаточным уровнем знаний и практики использования конкретного языка программирования. По языкам программирования большое преимущество имеет язык программирования Python, метод `split()` разделяет строки, в других языках такого метода нет, поэтому программа получается объемной и нужно писать достаточно много условий, не допуская ошибок.

Пути устранения типичных ошибочных ответов в ходе обучения школьников:

Больше отрабатывать на практике базовые приемы программирования со строчными данными, как в формате урочной работы, так и индивидуальной работы с учениками с поддержкой онлайн-ресурсов.

Проводить детальный анализ верных и неверных алгоритмов, акцентируя внимание на типичных ошибках.

Работать над метапредметными умениями - внимательно читать задание, четко анализировать условие, аккуратно оформлять и перепроверять решение.

Задание № 25

На умение создавать собственные программы (10–20 строк) для обработки целочисленной информации. Задание **высокого** уровня сложности.

Содержание задания:

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.

Решение:

Пример программы на C++:

```
#define _CRT_SECURE_NO_WARNINGS //25
#include <iostream>
#include <algorithm>
using namespace std;
int f(int n) {
    int m = n, k = 0;
    while (m != 0)
    {
        if (m % 10 == 9) k++;
        m /= 10;
    }
    return k;
}
int pr(int n) {
    int k = 0;
    for (int i = 2; i * i <= n; i++)
    {
        if (n % i == 0) k++;
        if (k > 0) break;
    }
    return k;
}
int g(int n, int& q) {
    int k = 0;
    for (int i = 2; i * i <= n; i++)
    {
        if (n % i == 0) {
            if (pr(i) == 0 and pr(n / i) == 0 and f(i) == 1 and f(n /
i) == 1)
            {
                k += 2; if (q < n / i) q = n / i; break;
            }
        }
    }
    return k;
}
int main() {
    int n, q, k = 0;
    for (n = 2920574; n <= 1e+8; n++) {
        q = 1;
        if (g(n, q) == 2) {
            k++; cout << n << ' ' << q << endl;
            if (k == 5) break;
        }
    }
    return 0;
}
```


}

Ответ:

2920661 153719

2920681 1709

2920781 8369

2920847 19603

2920849 6971

Типичные ошибки при выполнении:

- неумение создавать программы для обработки целочисленной информации (например, операции нахождения цифры числа);
- невнимательное прочтение условия заданий (например, последнее условие может быть понято как условие для самого числа, а не для делителей);
- создание программы с большим количеством циклов, программа работает очень долго, итоговые ответы могут быть не получены;
- незнание определения простого числа;
- неумение находить простые делители оптимальным способом;
- непонимание условия заданий и формализации требуемых условий.

Анализ возможных причин получения выявленных типичных ответов: Для решения заданий высокого уровня сложности требуется хорошая математическая подготовка, развитое логическое мышление в сочетании с достаточными навыками программирования. Основная ошибка в данном задании вызвана тем, что не учтены все условия; условий было достаточно много, больше, чем в прошлые годы, т.е. ошибка в алгоритме решения задачи. Также ошибки могут быть при написании программы в связи с недостаточным уровнем знаний и практики использования конкретного языка программирования.

Пути устранения типичных ошибочных ответов в ходе обучения школьников: Больше отрабатывать на практике базовые приемы программирования числовыми данными. Проводить детальный анализ верных и неверных алгоритмов, акцентируя внимание на типичных ошибках. Работать над метапредметными умениями - внимательно читать задание, четко анализировать условие, аккуратно оформлять и перепроверять решение.

Задание № 26

На умение обрабатывать целочисленную информацию с использованием сортировки.
Задание **высокого** уровня сложности.

Содержание задания:

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, имеющихся 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 ($N < 10000$) – 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:

Рассмотрим решение на C++:

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <vector>
#define int long long
#define fi first
#define se second
using namespace std;
signed main() {
    freopen("325_26.txt", "r", stdin);
    int n, d; cin >> n >> d;
    vector<pair<int, char>> a(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i].fi >> a[i].se;
    }
    sort(a.rbegin(), a.rend());

    vector<int> dp(n, 0);

    for (int i = 0; i < n; i++) {
        vector<int> fun;
        for (int j = 0; j < i; j++) {
            if (a[j].fi - a[i].fi >= d && a[i].se != a[j].se) {
                fun.push_back(dp[j]);
            }
        }
    }
}
```

```

        if (fun.size() > 0) {
            dp[i] = *max_element(fun.begin(), fun.end()) + 1;
        }
        else {
            dp[i] = 1;
        }
        fun.clear();
    }
    int q = *max_element(dp.begin(), dp.end());
    int max1 = -1;
    for (int i = 0; i < n; i++) {
        if (dp[i] == q) {
            max1 = max(max1, a[i].fi);
        }
    }
    cout << q << " " << max1;
    return 0;
}

```

Ответ: 166 63.

Типичные ошибки при выполнении:

- неверное применение методов сортировки в программах;
- неверное использование структур данных в программах (например, структур, пар, векторов, массивов);
- неумение создавать программы для обработки целочисленной информации;
- неумение создавать программы для реализации сложных математических алгоритмов;
- невнимательное прочтение условия заданий;
- решение заданий по шаблону, без учета изменений в формулировках условий и самих вопросов.
- непонимание условия заданий и формализации требуемых условий.

Анализ возможных причин получения выявленных типичных ответов: для решения заданий высокого уровня сложности требуется хорошая математическая подготовка, развитое логическое мышление в сочетании с достаточными навыками программирования. Для решения следует использовать конструкции пар, структуры данных, специальную сортировку с компаратором для данной структуры. При применении только пар программа была бы длиннее и нужно было бы использовать не одну пару векторов, а как минимум две пары. Не все учащиеся с ними знакомы, поскольку структуры данных и алгоритмы работы с ними относятся к алгоритмам решения олимпиадных задач по программированию и не всегда изучаются даже на уроках углубленного курса информатики. А также заметим, что файл, используемый в данном задании не имеет очень большого объема, поэтому задание можно решать с применением табличных технологий (специальных сумм, фильтрации и сортировкой данных).

Пути устранения типичных ошибочных ответов в ходе обучения школьников:

Изучение различных алгоритмов разработки, сортировки данных.

Освоение методик и схем, при помощи которых ученик находит новые способы решения задач, вырабатывает нестандартные планы достижения цели, оптимизирует ресурсы.

Поскольку заданный файл очень большой, то у ученика необходим навык проверки своего решения, он должен уметь создавать маленькие тесты (набор данных или текстовые файлы).

Внимательно читать условие, например, вместо суммы, находит среднее значение, а необходимые промежуточные данные были правильно найдены.

Задание № 27

Задание **высокого** уровня сложности. Оно направлено на 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.

Содержание задания:

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

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: R_x – среднее арифметическое абсцисс центров кластеров, и R_y – среднее арифметическое ординат центров кластеров.

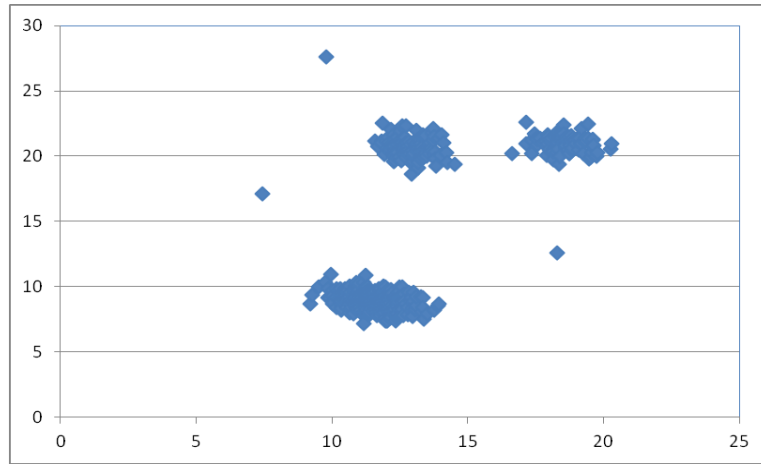
Для файла Б определите координаты центра каждого кластера, затем найдите два числа: Q_x – сумму абсцисс центров кластеров с минимальным и максимальным количеством точек, и Q_y – сумму ординат центров кластеров с минимальным и максимальным количеством точек.

Гарантируется, что во всех кластерах количество точек различно.

В ответе запишите четыре числа: в первой строке – сначала абсолютную величину целой части произведения $R_x \times 10\,000$, затем абсолютную величину целой части произведения $R_y \times 10\,000$; 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



Решение:

Используя электронную таблицу, можем сразу определить границы кластеров (визуально).



Откроем электронную таблицу для задачи А и построим точечную диаграмму. Видно, что здесь можно выделить два кластера, которые разделяются прямой $x = 5$. Все точки слева от этой прямой нужно отнести к кластеру 1, а все точки с $x > 5$ – к кластеру 2.

Рассмотрим решение на C++. Чтобы написать программу, текстовый файл нужно отредактировать, удалить первую строку и заменить все запятые на точки.

```
#define _CRT_SECURE_NO_WARNINGS //27
#include <iostream>
#include <algorithm>
#include <vector>
#include <iomanip>
using namespace std;
int main() {
    freopen("325_27_A.txt", "r", stdin);
    vector<pair<float, float>> a, b;
    pair<float, float> p;
    int n = 209;
    for (int i = 0; i < n; i++)
    {
        cin >> p.first >> p.second;
        if (p.first < 5) a.push_back(p); else b.push_back(p);
    }
}
```

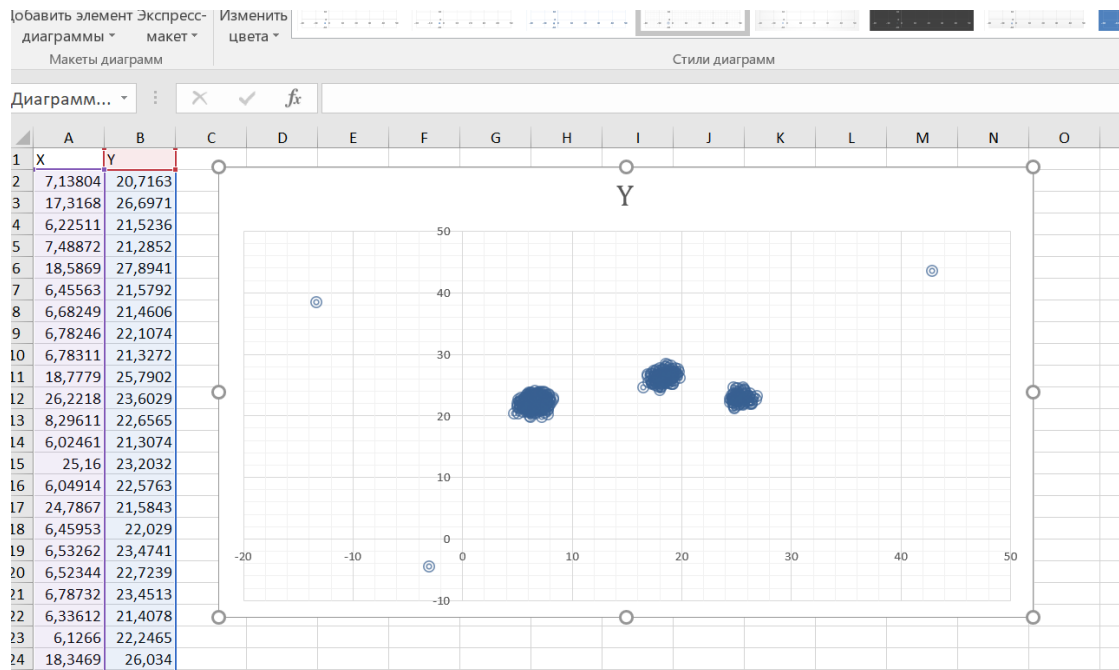
```

    float x1 = a[0].first, y1 = a[0].second, x2 = b[0].first, y2 =
b[0].second;
    int im1 = 0, im2 = 0;
    float smin = 1e+10;
    for (int i = 0; i < a.size(); i++)
    {
        float s = 0;
        for (int j = 0; j < a.size(); j++)
        {
            if (i != j) {
                s += sqrt((a[i].first - a[j].first) * (a[i].first -
a[j].first) + (a[i].second - a[j].second) * (a[i].second - a[j].second));
            }
            if (smin > s) { smin = s; im1 = i; }
        }
        smin = 1e+10;
        for (int i = 0; i < b.size(); i++)
        {
            float s = 0;
            for (int j = 0; j < b.size(); j++)
            {
                if (i != j)
                {
                    s += sqrt((b[i].first - b[j].first) * (b[i].first -
b[j].first) +
                        (b[i].second - b[j].second) * (b[i].second -
b[j].second));
                }
                if (smin > s) { smin = s; im2 = i; }
            }
        }
        cout << fixed << setprecision(7) << (a[im1].first + b[im2].first) /
2.0 * 10000 << ' '
        << (a[im1].second + b[im2].second) / 2.0 * 10000 << endl;
        return 0;
    }
}

```

Программа выводит 55295.1908112 136872.9972839. В ответе пишем целую часть чисел.

Таким же образом откроем электронную таблицу для задачи Б и построим точечную диаграмму. Заметим, что здесь можно выделить три кластера, которые разделяются прямыми $x = 10$ и $x = 22$.



Эти лишние три точки лишние удаляем вручную с текстового файла.

	A	B
1	X	Y
5	-3,0546	-4,5101
7	-13,285	38,4196
7	42,8679	43,4196

Рассмотрим решение на C++. Чтобы написать программу текстовый файл нужно отредактировать, удалить первую строку и заменить все запятые на точки.

```
#define _CRT_SECURE_NO_WARNINGS //27
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
int main() {
    freopen("325_27_B.txt", "r", stdin);
    vector<pair<float, float>> a, b, c;
    pair<float, float> p;
    int n = 610 - 1 - 3;
    for (int i = 0; i < n; i++)
    {
        cin >> p.first >> p.second;
        if (p.first > 22) c.push_back(p);
        else if (p.first < 10) a.push_back(p); else b.push_back(p);
    }
    float x1 = a[0].first, y1 = a[0].second, x2 = b[0].first, y2 =
b[0].second,
        x3 = c[0].first, y3 = c[0].second;
    int im1 = 0, im2 = 0, im3 = 0;

    float smin = 1e+10;
    for (int i = 0; i < a.size(); i++)
    {
        float s = 0;
        for (int j = 0; j < a.size(); j++)
        {
            if (i != j) {
```

```

        s += sqrt((a[i].first - a[j].first) * (a[i].first -
a[j].first) + (a[i].second - a[j].second) * (a[i].second - a[j].second));
    }
    } if (smin > s) { smin = s; im1 = i; }
}

smin = 1e+10;
for (int i = 0; i < b.size(); i++)
{
    float s = 0;
    for (int j = 0; j < b.size(); j++)
    {
        if (i != j)
        {
            s += sqrt((b[i].first - b[j].first) * (b[i].first -
b[j].first) +
                    (b[i].second - b[j].second) * (b[i].second -
b[j].second));
        }
    } if (smin > s) { smin = s; im2 = i; }
}

smin = 1e+10;
for (int i = 0; i < c.size(); i++)
{
    float s = 0;
    for (int j = 0; j < c.size(); j++)
    {
        if (i != j)
        {
            s += sqrt((c[i].first - c[j].first) * (c[i].first -
c[j].first) +
                    (c[i].second - c[j].second) * (c[i].second -
c[j].second));
        }
    } if (smin > s) { smin = s; im3 = i; }
}

int m1 = a.size(), m2 = b.size(), m3 = c.size(), xmax, xmin, ymax,
ymin;
cout << m1 << " " << m2 << " " << m3 << endl; //397 137 72

cout << (a[im1].first + c[im3].first) * 10000 << ' ' <<
(a[im1].second + c[im3].second) * 10000 << endl;
return 0;
}

```

Программа выводит 319185 448643.

Ответ:

А: 55295 136872

Б: 319185 448643

Типичные ошибки при выполнении:

- неумение работать с текстовыми файлами;
- неверное использование структур данных в программах (например, структур, пар, векторов, массивов);
- неумение создавать программы для реализации сложных математических алгоритмов;
- невнимательное прочтение условий заданий, возможны различные вариации ответов;

- решение заданий по шаблону, без учета изменений в формулировках условий и самих вопросов;
- получение целой части ответа в программе, зависит от компилятора, нужно быть очень внимательным, учитывать специфику языка программирования и типов данных в нем;
- непонимание условий заданий и формализации требуемых условий.

Анализ возможных причин получения выявленных типичных ответов: Для решения заданий высокого уровня сложности требуется хорошая математическая подготовка, развитое логическое мышление в сочетании с достаточными навыками программирования. Для решения следует использовать конструкции пар, структуры данных. Не все учащиеся с ними знакомы, поскольку структуры данных и алгоритмы работы с ними относятся к алгоритмам решения олимпиадных задач по программированию и не всегда изучаются даже на уроках углубленного курса информатики. Также заметим, что файл, используемый в данном задании может быть очень большого объема. Для языков программирования таких как С++ или Паскаль, текстовый файл нужно обрабатывать, заменяя запятые на точки, чтобы считывались дробные числа. Задание можно решать с применением табличных технологий (специальных сумм и фильтрацией/сортировкой данных), что требует лицензию табличного редактора. Возникли проблемы при получении точного ответа, нужно учитывать специфику типов данных в языке программирования. Компилятор может приблизить число не в пользу учащегося, например, в языке С++, так как числа задаются с точностью до 6-7 знаков после десятичной точки.

Пути устранения типичных ошибочных ответов в ходе обучения школьников: Изучение различных алгоритмов разработки данных. Освоение методик и схем, при помощи которых ученик находит новые способы решения задач, вырабатывает нестандартные планы достижения цели, оптимизирует ресурсы. Задание можно решить двумя способами: с помощью табличного редактора и программированием. Для решения данного задания с помощью табличного редактора он должен быть с обновлением или с лицензией, что не во всех образовательных организациях, к сожалению, есть, так как необходимо использовать функцию суммы для нахождения точки (центра кластера) с минимальным расстоянием от всех точек.

1.1.3. Анализ метапредметных результатов обучения, повлиявших на выполнение заданий КИМ

Согласно ФГОС СОО, должны быть достигнуты не только предметные, но и метапредметные результаты обучения.

Метапредметные результаты освоения основной образовательной программы должны, в частности, отражать овладение универсальными учебными познавательными действиями: познавательные УУД (базовые логические действия, базовые исследовательские действия, работа с информацией), коммуникативные УУД (общение), регулятивные УУД (самоорганизация, самоконтроль). Самостоятельно определять цели деятельности, выбирать успешные стратегии, владеть навыками познавательной, учебно-исследовательской и проектной деятельности, решать проблемы, критически оценивать и интерпретировать информацию, использовать средства информационных и коммуникационных технологий, владеть навыками познавательной рефлексии - все эти умения помогают обучающимся лучше справляться с различными типами задач и улучшают их общую подготовку в области информатики.

Основные метапредметные образовательные результаты, достигаемые в процессе подготовки школьников в области информатики:

- уверенная ориентация обучающихся в различных предметных областях при изучении школьных дисциплин, таких общепредметных понятий, как «объект», «система», «модель», «алгоритм», «исполнитель» и др.;

- владение основными общеучебными умениями информационно-логического характера: анализ объектов и ситуаций; синтез как составление целого из частей и самостоятельное достраивание недостающих компонентов; построение логических цепочек рассуждений и т.д.;

- владение основными универсальными умениями информационного характера: постановка и формулирование проблемы; поиск и выделение необходимой информации, применение методов информационного поиска; структурирование и визуализация информации; выбор наиболее эффективных способов решения задач в зависимости от конкретных условий; самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера;

- широкий спектр умений и навыков использования средств информационных и коммуникационных технологий для сбора, хранения, преобразования и передачи различных видов информации (работа с текстом, гипертекстом, звуком и графикой в среде соответствующих редакторов; создание и редактирование расчетных таблиц для автоматизации расчетов и визуализации числовой информации в среде табличных процессоров; хранение и обработка информации в базах данных; поиск, передача и размещение информации в компьютерных сетях), навыки создания личного информационного пространства.

Анализ метапредметных результатов обучения, влияющих на выполнение заданий КИМ, указывает на несколько ключевых областей, где обучающиеся испытывают трудности: во-первых, это слабая сформированность навыков смыслового чтения, которая привела к неправильной трактовке поставленной задачи, которые в свою очередь повлекли к неправильному составлению плана реализации намеченного пути решения, во-вторых, слабый уровень умения выбирать успешные стратегии в различных ситуациях, который проявился как неспособность разрабатывать, выстраивать, реализовывать алгоритмы для решения задач, в-третьих, способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания, включая работу с файлами, массивами, сортировку и обработку числовой и символьной информации.

Результаты свидетельствуют о необходимости усиления работы по улучшению метапредметных результатов посредством применения новых методов и подходов при обучении, а также использования метапредметных задач, которые доступны в открытом банке.

На выполнение влияют навыки самоорганизации и самоконтроля. Несформированность таких метапредметных результатов как самоорганизация и самоконтроль повлияла на низкую результативность экзамена в части перечисленных заданий. При анализе причин возникновения ошибок чаще всего встречалось банальное «невнимательное прочтение условия», которое приводило к неверному алгоритму решения и ошибкам. Многие экзаменуемые забывают вводить в установленные поля ответы к заданиям. Возможно высокий процент не приступивших к выполнению рассмотренных выше заданий в том числе зависит и от этого.

Недостаточно развитые метапредметные умения снизили баллы тех, кто ошибся при решении задач базового уровня № 5 и № 6 таких, как получение информации из источников разных типов, самостоятельно осуществлять анализ, систематизацию и интерпретацию информации различных видов и форм представления, оценивать достоверность информации, использовать средства информационных и коммуникационных технологий в решении задач. В задании № 5 участники в качестве ответа ищут не минимальный, а максимальный результат, или

результат в двоичной форме, а не в десятичной системе счисления, как сказано в задании, или исходное число, из которого получен нужный результат, а не само полученное значение. В задании № 6 – большое количество участников искали периметр объединения или количество точек пересечения областей, а не объединения как сказано в условии.

В задании № 8 требуется на высоком уровне проявить базовые логические действия: при поиске подходящих комбинаций цифр необходимо: устанавливать существенный признак классификации, основания для обобщения и сравнения, критерии проводимого анализа, выявлять и характеризовать существенные признаки объектов. Видим, что это задание выполняется участниками экзамена плохо.

Низкий процент выполнения задания № 9 прежде всего связан с несформированностью навыков формализации и моделирования, относящихся к навыкам работы с информацией. Спад указывает на невысокую сформированность универсальных познавательных действий: разрабатывать план решения проблемы с учётом анализа имеющихся ресурсов, способность и готовность к самостоятельному поиску методов решения практических задач, самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне.

Рекомендации для учителей: развивать алгоритмическое мышление, использовать задачи на исполнителей (Чертёжник, Робот, Черепаха в задании № 6) для отработки анализа алгоритмов, практиковать трассировку программ на бумаге, учить читать код до его выполнения (разбор программ "в уме"), давать задания на поиск ошибок в коде и их исправление в задании № 5, использовать визуализацию (деревья перебора, таблицы), разбирать задачи на построение слов по правилам (например, с ограничениями на позиции букв) в задании № 8, учить работать с Excel/Google таблицами (фильтры, формулы, условное форматирование), давать задачи на анализ реальных данных (например, статистика, результаты опросов), приучать учащихся к проверке решений (особенно в заданиях с перебором), разбирать типичные ошибки (например, пропуск вариантов в задании № 8), связывать информатику с математикой (комбинаторика, логика) и другими предметами (например, анализ данных из естественных наук).

Успешное выполнение заданий № 5, № 6, № 8, № 9 требует не только предметных знаний, но и развитых метапредметных навыков: логики, алгоритмического мышления, работы с данными и системного подхода. Учителям следует делать акцент на практику, анализ ошибок и междисциплинарные связи.

Навыки самоорганизации и самоконтроля, такие как самостоятельное планирование и осуществление целенаправленной деятельности, включая умения анализировать поставленную задачу и те условия, в которых она должна быть реализована; находить эффективные пути достижения результата; выявлять альтернативные, нестандартные способы решения познавательных задач; оценивать правильность выполнения поставленной познавательной задачи. Эти навыки особенно важны для выполнения компьютерных заданий всех уровней сложности, поскольку они, как правило, предполагают разбиение процесса выполнения заданий на несколько этапов, в каждом из которых требуется продемонстрировать владение как теоретическими, так и практико-ориентированными элементами содержания курса. При этом невнимательное прочтение формулировки задания, неверное выделение всех условий и неверное планирование своих действий может привести к неверному ответу и (или) неэффективному выполнению задания с точки зрения временных затрат.

Именно такими заданиями являются задания № 17, №№ 24-27.

В решении заданий № 17, №№ 24-27 учащийся должен выполнить следующие этапы:

- анализ условия задачи, пример организации входных и выходных данных;

- уметь правильно считывать данные (числовые, строковые) из текстового файла;
- графическое изображение возможной схемы расположения данных соотнесение его с форматом входных данных; для задания Б определить точки вне кластеров – для задания № 27;
- формулирование переборного алгоритма или использование специальных структур, таких как векторы, множества, его отладка;
- самопроверка решения, а также важна проверка как на собственных маленьких тестах;
- внесение ответов, вывод ответов по правильному формату.

Особенностью компьютерного ЕГЭ является вариативность решения большинства задач: при этом можно использовать аналитический метод, решение средствами электронных таблиц или написание программы. Нет четких рекомендаций, какое из заданий решается каким способом. Цель учителя - познакомить учеников с вариантами решения, а выбор и его реализация остаются за участниками экзамена. Также метапредметным результатом является способность правильно распределить время экзамена между заданиями для получения максимального количества баллов.

Задания № 9, № 17 и № 24 - 27, когда ученик должен сначала разработать алгоритм поиска верного решения, а затем его реализовать средствами электронных таблиц или программированием, тесно связаны со второй и третьей группами метапредметных результатов, в которых проверяется способность и готовность к успешной самостоятельной работе. К сожалению, процент выполнения большинства этих заданий упал, что говорит о недостаточной сформированности метапредметных результатов обучения, отвечающих за познавательную активность учеников и способность выбора эффективного метода решения задачи.

Рекомендации для учителей: развитие алгоритмического мышления, больше практики разбора сложных задач (например, разбор олимпиадных задач), обучение декомпозиции – разбивать задачу на подзадачи, анализ нескольких решений одной задачи (например, от полного перебора до оптимизированного эффективного алгоритма), тренировка обработки больших данных (файлы, массивы, строки), обучение методам сокращения перебора (например, в задачах на делители), сделать акцент на эффективных структурах данных (хеш-таблицы, множества). Развитие отладки и тестирования, привычка проверять крайние случаи (нули, пустые строки, максимальные значения), разбор типовых ошибок (переполнение, выход за границы массива), постепенное усложнение задач на ДП (от простых к сложным), визуализация решений (таблицы, графы), разбор классических задач (кузнечик, рюкзак, наибольшая подпоследовательность). Формирование вычислительного мышления, решение задач с ограничениями по времени (тренировка скорости), анализ сложности алгоритмов ($O(n)$, $O(n \log n)$ и т.д.) и т.п.

Успешное выполнение заданий № 17, № 24 - № 27 требует не только предметных знаний, но и развитых метапредметных навыков: алгоритмического мышления, умения оптимизировать решения, работы с данными, тестирования кода. Учителям стоит делать упор на практику, разбор ошибок и постепенное усложнение задач.

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***

Модель ЕГЭ-2025 по информатике стала естественным продолжением модели прошлых лет. В работу заменено задание № 27, проверяющее умение выполнять последовательность действий при анализе данных: сбор первичных данных, очистка и оценка качества данных, выбор и построение модели, преобразование данных, визуализация данных, интерпретация результатов.

Итак, результаты ЕГЭ текущего года указывают на то, что следующие элементы содержания обучающимися усвоены на достаточном уровне:

Задания базового уровня:

Задание № 1. Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы) (84,4%);

Задание № 2. Умение строить таблицы истинности и логические схемы (65,8%);

Задание № 3. Умение поиска информации в реляционных базах данных (69,5%);

Задание № 4. Умение кодировать и декодировать информацию (81,1%);

Задание № 7. Умение определять объём памяти, необходимый для хранения графической и звуковой информации (51,2%);

Задание № 10. Информационный поиск средствами текстового процессора (69,8%);

Задание № 19. Умение анализировать алгоритм логической игры (61,3%).

Задания повышенного и высокого уровня

Задание № 11. Умение подсчитывать информационный объём сообщения (19,9%);

Задание № 12. Умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд (28,3%);

Задание № 13. Умение использовать маску подсети (29,3%);

Задание № 14. Знание позиционных систем счисления (30,4%);

Задание № 15. Знание основных понятий и законов математической логики (35,5%);

Задание № 16. Вычисление рекуррентных выражений (39,5%);

Задание № 18. Умение использовать электронные таблицы для обработки целочисленных данных (29,6%);

Задание № 20. Умение найти выигрышную стратегию игры (37,8%);

Задание № 21. Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию (36,1%);

Задание № 22. Построение математических моделей для решения практических задач. Архитектура современных компьютеров. Многопроцессорные системы (25,3%);

Задание № 23. Умение анализировать ход исполнения алгоритма (33,9%).

- ***Перечень элементов содержания / умений и видов деятельности, усвоение которых всеми школьниками региона в целом, школьниками с разным уровнем подготовки нельзя считать достаточным***

В этом году наибольшую трудность для обучающихся составили задания высокого уровня № 24 - 27, что говорит о недостаточной сформированности у обучающихся умения создавать собственные программы (20–40 строк) для анализа строковых и числовых последовательностей, а также целочисленной арифметики.

Недостаточно усвоены следующие элементы содержания, проверяемые на ЕГЭ:

Задания базового уровня:

Задание № 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 (28,8%).

Задание № 6. Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов (40,3%).

Задание № 8. Знание основных понятий и методов, используемых при измерении количества информации (34,4%).

Задание № 9. Умение обрабатывать числовую информацию в электронных таблицах (19,8%).

Задания повышенного и высокого уровня

Задание № 17. Умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования (13,7%).

Задание № 24. Умение создавать собственные программы (10–20 строк) для обработки символьной информации (2,2%).

Задание № 25. Умение создавать собственные программы (10–20 строк) для обработки целочисленной информации (4,1%).

Задание № 26. Умение обрабатывать целочисленную информацию с использованием сортировки (3,1%).

Задание № 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 (6,7%).

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

В 2025 году в заданиях базового уровня №№ 2, 5, 19 существенно понизился процент выполнения по сравнению с 2024 г., что показывает недостаточную сформированность умений строить таблицы истинности и логические схемы, строить простой алгоритм, записанный на естественном языке, или умение создавать линейный алгоритм, анализировать алгоритм логической игры. В заданиях повышенного и высокого уровня №№ 12, 16 - 21, 23, 25 выявлены проблемы с умением исполнить алгоритм для конкретного исполнителя с фиксированным набором команд, вычислением рекуррентных выражений, умением составить алгоритм обработки числовой последовательности, использованием электронных таблиц для обработки целочисленных данных, поиском выигрышной стратегии игры, построением дерева игры по заданному алгоритму и поиском выигрышной стратегии, анализом хода исполнения алгоритма, созданием собственных программ для обработки целочисленной информации.

Очевидно, что большинство экзаменуемых в достаточной степени усваивают темы, необходимые для решения заданий базового уровня, но изучают эти темы недостаточно глубоко, что влияет на выполнение заданий высокого уровня сложности. Кроме того, у экзаменуемых возникают проблемы с заданиями, при выполнении которых требуются не только знания основных элементов языка программирования, но и умение составлять и создавать собственные программы.

При подготовке к экзамену необходимо уделять внимание заданиям, связанным с более глубоким анализом различных ситуаций (в том числе на основе стандартных алгоритмов),

элементам программирования, умению анализировать готовую программу и создавать собственную программу.

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

Все мероприятия, включенные в дорожную карту в 2025 году, были реализованы. Однако в текущем году факт снижения показателей говорит о недостаточной эффективности проведенной работы. В связи с этим в наступающем учебном году с целью повышения качества обучения и подготовки к экзаменам необходимо на основе детального анализа результатов ЕГЭ выстроить продуманную системную методическую работу со всеми ОО, оказать адресную помощь учителям, испытывающим трудности в методическом и предметном плане.