



**ДЕПАРТАМЕНТ
ОБРАЗОВАНИЯ
ТОМСКОЙ ОБЛАСТИ**

Фрунзе пр., 14, г. Томск, 634029
тел. (382 2) 467-900
E-mail: prm@do.tomsk.gov.ru

ИНН/КПП 7021019573/701701001, ОГРН 1027000863670

18.07.2024 № 65-6689

на № _____ от _____

О направлении методических рекомендаций «Особенности подготовки к ГИА по учебному предмету «Информатика»»

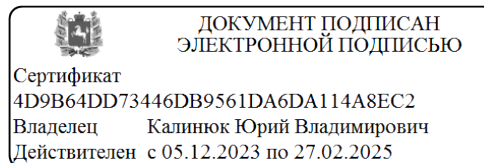
Руководителям органов местного самоуправления, осуществляющих управление в сфере образования Томской области

Уважаемые руководители!

Департамент образования Томской области направляет для использования в работе методические рекомендации «Особенности подготовки к государственной итоговой аттестации по учебному предмету «Информатика»».

Приложение на 48 л. в 1 экз.

Начальник департамента



Ю.В. Калинин



ТО-21654218

Замятина Оксана Михайловна
8 (382 2) 90 79 89
toipkro@toipkro.ru

Методические рекомендации
«Особенности подготовки к государственной итоговой аттестации по учебному предмету «Информатика»»

Составитель:

Сайфутдинова Д.В., старший преподаватель центра развития педагогического мастерства ТОИПКРО

Государственная итоговая аттестация (ГИА) по информатике проводится в соответствии с Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» и является экзаменом по выбору. ГИА по информатике по образовательным программам среднего общего образования проходит в форме Единого государственного экзамена (ЕГЭ) для обучающихся, не имеющих академической задолженности и в полном объеме выполнивших учебный план по программам среднего общего образования. Государственная итоговая аттестация выпускников 11 классов нацелена на проверку предметных знаний и компетенций, заявленных в федеральном государственном образовательном стандарте (ФГОС).

Настоящие методические рекомендации нацелены на оказание методической помощи учителям общеобразовательных организаций Томской области, участвующих в подготовке школьников к государственной итоговой аттестации в форме единого государственного экзамена по информатике.

Краткая характеристика КИМ ЕГЭ по информатике

В 2023 году ЕГЭ по предмету «Информатика» третий раз проводился в компьютерной форме (КЕГЭ¹). Каждому участнику экзамена предоставляется автоматизированное рабочее место без выхода в сеть «Интернет» с установленным специализированным программным обеспечением (ПО) «Станция КЕГЭ»², набором стандартного ПО (текстовые редакторы, редакторы электронных таблиц, среды программирования на языках: Школьный алгоритмический язык, C#, C++, Pascal, Java, Python).

В 2023 году существенных изменений в модели экзамена не произошло, за исключением замены двух заданий (6 и 22). Задание 22 повышенного уровня сложности выполнялось с использованием файла, содержащего модель системы с параллельными процессами. В задании 6 базового уровня анализ алгоритма выполнялся на примере фрагмента программы для исполнителя «Черепашка».

Для выполнения части заданий (11 из 27) требовался компьютер, оснащенный специальным программным обеспечением, поэтому от учащихся требовалось практическое владение соответствующими средствами. В число этих 11 заданий входили задания на практическое программирование, работу с электронными таблицами и базой данных, а также информационный поиск средствами текстового редактора.

¹ КЕГЭ - ЕГЭ по учебному предмету «Информатика», проводимый в компьютерной форме.

² ПО «Станция КЕГЭ» - Специализированное программное обеспечение, устанавливаемое на компьютерах (ноутбуках), за которыми участники КЕГЭ проходят экзамен.

Согласно спецификации КИМ⁵ ЕГЭ 27 заданий были распределены по следующим основным тематическим блокам курса информатики:

- информация и её кодирование;
- моделирование и компьютерный эксперимент;
- системы счисления;
- логика и алгоритмы;
- элементы теории алгоритмов;
- программирование;
- обработка числовой информации;
- технологии поиска и хранения информации.

Обобщённый план варианта КИМ ЕГЭ 2023 года по информатике представлен ниже.

Задание 1 базового уровня сложности проверяет умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы), на работу с которым не требуется использование специализированного ПО.

Задание 2 базового уровня сложности проверяет умение строить таблицы истинности и логические схемы без использования специального ПО.

Задание 3 базового уровня сложности проверяет умение поиска информации в реляционных базах данных с использованием специального ПО (работа с реляционными базами данных).

Задание 4 базового уровня сложности проверяет умение кодировать и декодировать информацию без использования специального ПО.

Задание 5 базового уровня сложности 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 без использования специального ПО.

Задание 6 базового уровня сложности проверяет получение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов без использования специального ПО.

Задание 7 базового уровня сложности проверяет умение определять объем памяти, необходимый для хранения графической и звуковой информации без использования специального ПО.

Задание 8 базового уровня сложности проверяет знание основных понятий и методов, используемых при измерении количества информации без использования специального ПО.

Задание 9 базового уровня сложности проверяет умение обрабатывать числовую информацию в электронных таблицах с использованием специального ПО (работа с электронными таблицами). Хотя это задание отнесено к базовому уровню сложности, но на самом деле его сложность очень высокая.

Задание 10 базового уровня сложности проверяет умение выполнять информационный поиск средствами операционной системы или текстового процессора с использованием специального ПО (текстовый процессор).

Задание 11 повышенного уровня сложности проверяет умение подсчитывать информационный объем сообщения без использования специального ПО.

Задание 12 повышенного уровня сложности проверяет умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд без использования специального ПО.

⁵ КИМ - контрольные измерительные материалы, по которым проводится государственная итоговая аттестация, представляющие собой комплексы заданий стандартизированной формы.

Задание 13 повышенного уровня сложности проверяет умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы) без использования специального ПО.

Задание 14 повышенного уровня сложности проверяет знание позиционных систем счисления без использования специального ПО.

Задание 15 повышенного уровня сложности проверяет знание основных понятий и законов математической логики без использования специального ПО.

Задание 16 повышенного уровня сложности проверяет умение проводить вычисление рекуррентных выражений с использованием специального ПО (электронные таблицы или среда программирования).

Задание 17 повышенного уровня сложности проверяет умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования с использованием специального ПО.

Задание 18 повышенного уровня сложности проверяет умение использовать электронные таблицы для обработки целочисленных данных с использованием специального ПО (работа с электронными таблицами).

Задания 19–21 на умение анализировать алгоритм логической игры, поиск выигрышной стратегии. Задание 19 базового уровня сложности проверяет умение анализировать алгоритм логической игры без использования специального ПО. Задание 20 повышенного уровня сложности проверяет умение найти выигрышную стратегию игры без использования специального ПО. Задание 21 высокого уровня сложности проверяет умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию без использования специального ПО.

Задание 22 повышенного уровня сложности проверяет умение построения математических моделей для решения практических задач, а также знание архитектуры современных компьютеров и многопроцессорных систем с использованием специального ПО.

Задание 23 повышенного уровня сложности проверяет умение анализировать результат исполнения алгоритма, содержащего ветвление и цикл без использования специального ПО.

Задание 24 высокого уровня сложности проверяет умение создавать собственные программы (10–20 строк) для обработки символьной информации с использованием специального ПО.

Задание 25 высокого уровня сложности проверяет умение создавать собственные программы (10–20 строк) для обработки целочисленной информации с использованием специального ПО.

Задание 26 высокого уровня сложности проверяет умение обрабатывать целочисленную информацию с использованием сортировки с использованием специального ПО.

Задание 27 высокого уровня сложности проверяет умение создавать собственные программы (20–40 строк) для анализа числовых последовательностей с использованием специального ПО.

Задания 1 и 13 относятся к математическому моделированию, в которых используются простейшие понятия теории графов, задания 2, 15, 19, 20, 21 относятся к математической логике и логическому анализу игры, задания 3, 9, 10, 18 относятся к базам данных, электронным таблицам и текстовым процессорам, задания 4, 7, 8, 11, 14 относятся к кодированию информации, задания 5, 6, 12, 16, 17, 22, 23, 24, 25, 26, 27 относятся к алгоритмизации и программированию.

Задания с 1 по 25 оценивались одним баллом, задания 26 и 27 – двумя баллами. Таким образом, общее количество первичных баллов равно 29, которые затем переводятся в тестовые баллы по 100-балльной шкале.

Так как в 2023 году ЕГЭ по предмету «Информатика» третий раз проводился в компьютерной форме, был исключен этап ручной проверки части заданий экспертами, что повысило объективность оценок. Однако переход к компьютерной форме сдачи экзамена, никак не повлиял на планомерное увеличение количества участников экзамена по информатике. Это обусловлено высокой востребованностью предмета как вступительного испытания для поступления в вузы.

Содержательный анализ выполнения заданий КИМ ЕГЭ по информатике

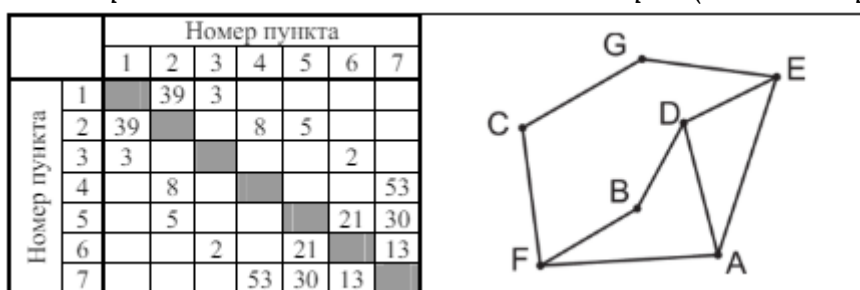
Содержательный анализ выполнения заданий КИМ проводится с учетом полученных результатов статистического анализа всего массива результатов экзамена по учебному предмету вне зависимости от выполненного участником экзамена варианта КИМ. Примеры заданий будут рассмотрены на примере обобщённого варианта КИМ, находящегося в открытом доступе на сайте ФИПИ (<https://fipi.ru/ege/demoversii-specifikacii-kodifikatory#!tab/151883967-5>).

Рассмотрим по порядку задания базового, повышенного и высокого уровня сложности.

Задание 1 относится к математическому моделированию, в котором используются простейшие понятия теории графов, с помощью которого проверяют материал, изучаемый как в основной, так и в средней школе. Тип задания относится к базовому уровню. Ошибки совершаются обучающимися из-за невнимательности или спешки.

Пример задания⁴.

На рисунке схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о протяжённости каждой из этих дорог (в километрах).



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова сумма протяжённости дорог из пункта D в пункт B и из пункта F в пункт A.

В ответе запишите целое число.

Для решения данного задания необходимо уметь анализировать пути на графе и подсчитывать их количество, уметь интерпретировать результаты, получаемые в ходе моделирования реальных процессов и соотносить их. Основные понятия, которые должен знать обучающийся для выполнения задания: граф (взвешенный граф), весовая матрица.

Задание 2 относится к заданиям математической логики. Для его решения необходимо уметь анализировать логические выражения и строить таблицу истинности.

⁴ Примеры заданий (здесь и далее) из демонстрационного КИМ, который не исчерпывает всего многообразия возможных формулировок заданий на каждой позиции варианта экзаменационной работы и методов их решения.

Пример задания.

Миша заполнял таблицу истинности логической функции F

$$\neg(y \rightarrow x) \vee (z \rightarrow w) \vee \neg z,$$

но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

				F
	0			0
0	1			0
1			0	0

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Функция F задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имеет следующий вид.

		F
0	1	0

В этом случае первому столбцу соответствует переменная y , а второму столбцу – переменная x . В ответе следует написать: yx .

Задание сводится к построению таблицы истинности для определенного значения функции (0 или 1) и сопоставление ее с фрагментом. Основные логические операции и приоритеты логических операций можно найти в инструкции к КИМ (в КЕГЭ в черновике).

Согласно спецификации, задание выполняется без специального ПО, то есть таблица истинности может быть построена вручную. Так же можно воспользоваться редактором электронных таблиц для построения таблицы истинности, используя логические функции ИЛИ(), И(), НЕ() и ЕСЛИ() или воспользоваться средой программирования и написать программу на выбранном языке (для вычисления значения функции необходимо понимать, как логические операторы записываются на языке программирования).

	A	B	C	D	E	F	G	H	I	J
1	y	x	z	w	y*не(x)	не(z)	не(z)+w	f		
2	0	0	1	0	ложь	ложь	ложь	ложь		
3	0	1	1	0	ложь	ложь	ложь	ложь		
4	1	1	1	0	ложь	ложь	ложь	ложь	$\neg(y \rightarrow x) \vee (z \rightarrow w) \vee \neg z$	
5	1	0	1	0	ИСТИНА	ложь	ложь	ИСТИНА	$\text{не}(\text{не}(y)+x)+(\text{не}(z)+w)+\text{не}(z)$	
6	0	0	0	0	ложь	ИСТИНА	ИСТИНА	ИСТИНА	$(y*\text{не}(x))+(\text{не}(z)+w)+\text{не}(z)$	
7	0	1	0	0	ложь	ИСТИНА	ИСТИНА	ИСТИНА		
8	1	0	0	0	ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА		
9	1	1	0	0	ложь	ИСТИНА	ИСТИНА	ИСТИНА		
10	0	0	0	1	ложь	ИСТИНА	ИСТИНА	ИСТИНА		
11	0	0	1	1	ложь	ложь	ИСТИНА	ИСТИНА		
12	0	1	0	1	ложь	ИСТИНА	ИСТИНА	ИСТИНА		
13	0	1	1	1	ложь	ложь	ИСТИНА	ИСТИНА		
14	1	0	0	1	ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА		
15	1	0	1	1	ИСТИНА	ложь	ИСТИНА	ИСТИНА		
16	1	1	1	1	ложь	ложь	ИСТИНА	ИСТИНА		

Рисунок 1. Построение таблицы истинности с помощью электронных таблиц

```

print('y x z w')
for y in 0, 1:
    for x in 0, 1:
        for z in 0, 1:
            for w in 0, 1:
                F = not(not(y) or x) or (not(z) or w) or not(z)
                if F==0:
                    print(y, x, z, w, F)

```

Рисунок 2. Построение таблицы истинности с помощью Python

Для решения задания 2 можно воспользоваться полностью программным способом решения этой задачи, пример таких решений можно найти на сайте А.К. Полякова (<https://kpolyakov.spb.ru/school/ege.htm>). Применение таких методов целесообразно при высоком уровне подготовки обучающегося, но могут быть трудозатратны по времени.

Решение задания 3 базового уровня сложности проверяет умение поиска информации в реляционных базах данных с использованием специального ПО (редактор электронных таблиц), с которым участники экзамена справляются хорошо. Для выполнения задания согласно кодификатору на базовом уровне обучающийся должен владеть навыками создания, ведения и использования баз данных при решении учебных и практических задач.

Пример задания.

В файле приведён фрагмент базы данных «Продукты» о поставках товаров в магазины районов города. База данных состоит из трёх таблиц.

Таблица «Движение товаров» содержит записи о поставках товаров в магазины в течение первой декады июня 2021 г., а также информацию о проданных товарах. Поле Тип операции содержит значение Поступление или Продажа, а в соответствующее поле Количество упаковок внесена информация о том, сколько упаковок товара поступило в магазин или было продано в течение дня. Заголовок таблицы имеет следующий вид.

ID операции	Дата	ID магазина	Артикул	Тип операции	Количество упаковок	Цена
-------------	------	-------------	---------	--------------	---------------------	------

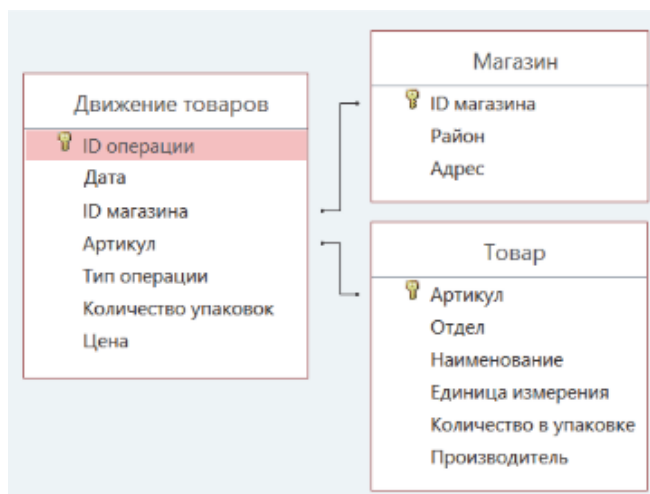
Таблица «Товар» содержит информацию об основных характеристиках каждого товара. Заголовок таблицы имеет следующий вид.

Артикул	Отдел	Наименование	Единица измерения	Количество в упаковке	Производитель
---------	-------	--------------	-------------------	-----------------------	---------------

Таблица «Магазин» содержит информацию о местонахождении магазинов. Заголовок таблицы имеет следующий вид.

ID магазина	Район	Адрес
-------------	-------	-------

На рисунке приведена схема указанной базы данных.



Используя информацию из приведённой базы данных, определите общий вес (в кг) крахмала картофельного, поступившего в магазины Заречного района за период с 1 по 8 июня включительно. В ответе запишите только число

Перед выполнением задания проводится анализ представленной базы данных и запроса для формирования этапов работы. Для нахождения результата можно редактировать базу данных по совокупности условий с помощью Сортировки и Фильтра, а после этого произвести арифметические операции с данными отобранных записей. Второй способ выполнения задания – это использование функции ВПР (поисковая функция в Excel) (искомое_значение; таблица; номер_столбца; интервальный_просмотр) и, при необходимости, проведение дополнительных вычислений. Функция просматривает содержимое ячеек в заданном интервале одной таблицы, а когда находит совпадение, переносит значение на вторую.

ID операции	Дата	ID магазина	Артикул	Количество упаковок, шт	Тип операции	Цена руб./шт.	Район	Наименование товара
1	01.06.2021	M1	4	180	Поступление	75	Октябрьский	Кефир 3,2%
2	01.06.2021	M1	4	180	Продажа	75	Октябрьский	Кефир 3,2%
3	01.06.2021	M1	5	180	Поступление	70	Октябрьский	Кефир обезжиренный
4	01.06.2021	M1	5	170	Продажа	70	Октябрьский	Кефир обезжиренный
5	01.06.2021	M1	6	180	Поступление	50	Октябрьский	Ряженка термостатная
6	01.06.2021	M1	6	180	Продажа	50	Октябрьский	Ряженка термостатная
7	01.06.2021	M1	9	180	Поступление	55	Октябрьский	Сметана 15%
8	01.06.2021	M1	9	150	Продажа	55	Октябрьский	Сметана 15%
9	01.06.2021	M1	10	180	Поступление	70	Октябрьский	Сметана 25%
10	01.06.2021	M1	10	150	Продажа	70	Октябрьский	Сметана 25%
11	01.06.2021	M1	13	170	Поступление	60	Октябрьский	Творог 9% жирности
12	01.06.2021	M1	13	120	Продажа	60	Октябрьский	Творог 9% жирности
13	01.06.2021	M1	18	180	Поступление	49	Октябрьский	Крупа манная
14	01.06.2021	M1	18	80	Продажа	49	Октябрьский	Крупа манная
15	01.06.2021	M1	24	180	Поступление	50	Октябрьский	Макароны спагетти
16	01.06.2021	M1	24	159	Продажа	50	Октябрьский	Макароны спагетти

Рисунок 3. Пример использования функции ВПР()

Задание 4 базового уровня сложности входит в список заданий по кодированию информации и проверяет умение кодировать и декодировать информацию без использования специального ПО.

Пример задания.

По каналу связи передаются сообщения, содержащие только буквы из набора: А, З, К, Н, Ч. Для передачи используется двоичный код, удовлетворяющий прямому условию Фано, согласно которому никакое кодовое слово не является началом другого кодового слова. Это условие обеспечивает возможность однозначной расшифровки закодированных сообщений. Кодовые слова для некоторых букв известны: Н – 1111, З – 110. Для трёх оставшихся букв А, К и Ч кодовые слова неизвестны. Какое количество

двоичных знаков потребуется для кодирования слова КАЗАЧКА, если известно, что оно закодировано минимально возможным количеством двоичных знаков?

Решение такого типа задач, в которых код букв должен удовлетворять условию Фано (т.е. однозначно декодироваться), сводится к построению бинарного дерева и его анализа. Дерево для двоичного кодирования начинается с двух направлений, которые означают 0(ноль) и 1(единицу) (цифры двоичного кодирования). От каждого направления можно рисовать только два направления: 0 (ноль) и 1(единицу) и т.д. Для удобства можно располагать 1(единицу) только вправо, а 0 (ноль) только влево. В конце каждой ветки можно располагать букву, которую мы хотим закодировать, но, если мы расположили букву, от этой ветки больше нельзя делать новых ответвлений. Например, расположим буквы, которые уже имеют код (Н, З), на бинарное дерево, и проанализировав оставшиеся ветки и повтор букв в слове, расположим оставшиеся буквы на дереве так, чтобы слово было закодировано минимальным количеством двоичных знаков. Такой подход позволяет однозначно декодировать сообщение, состоящее из этих букв.

Задания могут формулироваться по-разному, но принцип везде одинаков: с помощью бинарного дерева можно подобрать требуемые кодовые слова. Необходимо помнить, если в задании упомянут символ, который не используется в кодируемом слове, его всё равно нужно располагать в каком-нибудь листе дерева.

Задание 5 базового уровня сложности 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 без использования специального ПО. Задание относится к алгоритмизации и программированию.

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

1. Строится двоичная запись числа N .

2. Далее эта запись обрабатывается по следующему правилу:

а) если сумма цифр в двоичной записи числа чётная, то к этой записи справа дописывается 0, а затем два левых разряда заменяются на 10;

б) если сумма цифр в двоичной записи числа нечётная, то к этой записи справа дописывается 1, а затем два левых разряда заменяются на 11.

Полученная таким образом запись является двоичной записью искомого числа R .

Например, для исходного числа $6_{10} = 110_2$ результатом является число $1000_2 = 8_{10}$, а для исходного числа $4_{10} = 100_2$ результатом является число $1101_2 = 13_{10}$.

Укажите минимальное число N , после обработки которого с помощью этого алгоритма получается число R , большее 40. В ответе запишите это число в десятичной системе счисления.

Данный тип заданий может выполняться с использованием разных способов:

1) теоретическое решение задания, используя понятие бита четности, зная, что при добавлении к двоичной записи числа нуля справа, число увеличивается в 2 раза и при отбрасывании последней цифры в двоичной записи, число уменьшается в 2 раза (остаток от деления на 2 отбрасывается). Для выполнения вычислений можно использовать, например, Калькулятор в режиме Программист, консоль Python или электронные таблицы.

2) решение задания методом перебора через написание программы в соответствии с заданием. Данное решение можно реализовать в электронных таблицах с использованием функций перевода: =ДЕС.В.ДВ (число;разряд) и =ДВ.В.ДЕС (число) или с помощью написания программы на выбранном языке программирования.

Назначение	Функция
Для перевода числа N из десятичной в двоичную систему счисления	=ДЕС.В.ДВ(A2)
Для нахождения суммы цифр числа функция	=СУММПРОИЗВ(--ПСТР(B2;СТРОКА(ДВССЫЛ("A1:A"&ДЛСТР(B2)))));1))
Для проверки на четность	=ЕЧЁТН(C2)
Для добавления бита четности	=ЕСЛИ(D2;СЦЕПИТЬ(B2;0);СЦЕПИТЬ(B2;1))
Для замены первых 2 символов	=ЕСЛИ(D2;ЗАМЕНИТЬ(E2;1;2;"10");ЗАМЕНИТЬ(E2;1;2;"11"))
Для перевода числа из двоичной в десятичную в систему счисления	=ДВ.В.ДЕС(F2)
Поиск числа N, для которого больше R>40	Числовой фильтр (столбец G больше 40)

	A	B	C	D	E	F	G
	N	N2	Сумма цифр	Четность	Добавление бита четности	Замена первых символов	R
1	1	1	1	ЛОЖЬ	11	11	3
2	2	10	1	ЛОЖЬ	101	111	7
3	3	11	2	ИСТИНА	110	100	4
4	4	100	1	ЛОЖЬ	1001	1101	13
5	5	101	2	ИСТИНА	1010	1010	10
6	6	110	2	ИСТИНА	1100	1000	8
7	7	111	3	ЛОЖЬ	1111	1111	15
8	8	1000	1	ЛОЖЬ	10001	11001	25
9	9	1001	2	ИСТИНА	10010	10010	18
10	10	1010	2	ИСТИНА	10100	10100	20
11	11	1011	3	ЛОЖЬ	10111	11111	31
12	12	1100	2	ИСТИНА	11000	10000	16
13	13	1101	3	ЛОЖЬ	11011	11011	27
14	14	1110	3	ЛОЖЬ	11101	11101	29
15	15	1111	4	ИСТИНА	11110	10110	22
16	16	10000	1	ЛОЖЬ	100001	110001	49
17	17	10001	2	ИСТИНА	100010	100010	34
18	18	10010	2	ИСТИНА	100100	100100	36
19	19	10011	3	ЛОЖЬ	100111	110111	55
20	20	10100	2	ИСТИНА	101000	101000	40
21	21	10101	3	ЛОЖЬ	101011	111011	59
22	22	10110	3	ЛОЖЬ	101101	111101	61
23	23	10111	4	ИСТИНА	101110	101110	46
24	24	11000	2	ИСТИНА	110000	100000	32
25	25	11001	3	ЛОЖЬ	110011	110011	51
26	26	11010	3	ЛОЖЬ	110101	110101	53
27	27	11011	4	ИСТИНА	110110	100110	38
28	28	11100	3	ЛОЖЬ	111001	111001	57
29	29	11101	4	ИСТИНА	111010	101010	42
30	30	11110	4	ИСТИНА	111100	101100	44
31	31	11111	5	ЛОЖЬ	111111	111111	63
32	32	100000	1	ЛОЖЬ	1000001	1100001	97
33	33	100001	2	ИСТИНА	1000010	1000010	66
34	34	100010	2	ИСТИНА	1000100	1000100	68

Рисунок 4. Использование таблицы в методе перебора

```

16 49
19 55
21 59
22 61
23 46
25 51
for n in range(1,40):
    s=bin(n)[2:]
    if s.count('1')%2==0:
        r='10'+s[2:]+ '0'
    else:
        r='11'+s[2:]+ '1'
    r=int(r, 2)
    if r>40:
        print(n, r)
36 72
37 107
38 109
39 78

```

Рисунок 5. Программа на Python для решения задания 5

Типов данного задания, как и способов решения достаточно много, поэтому могут встретиться усложненные варианты задания. Например, экзаменуемым может потребоваться перевести число в троичную систему счисления. При подготовке важно изучить теорию для выполнения этого номера и тогда, полученные знания станут базой для решения упражнений с помощью того метода, который будет удобнее.

В 2023 году продолжается корректировка экзаменационных моделей по большинству учебных предметов в соответствии с ФГОС, поэтому задание 6 в 2023 году было посвящено анализу алгоритма для конкретного исполнителя, определению возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Данное задание относится к алгоритмизации и программированию, но по сравнению с другими из этой группы имеет минимальную решаемость – 22,16%, так как его очень трудно решить «вручную» на листе бумаги. Проще всего его решить в такой системе программирования, в которой есть функции исполнителя «черепашка». Это система КуМир или Python с соответствующей библиотекой. Решение состоит в том, что вначале черепашка рисует геометрическую фигуру из отрезков прямых, а затем можно «вручную» на рисунке подсчитать число точек с целочисленными координатами, попавшими внутрь фигуры.

Пример 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 известно положение исполнителя и направление его движения. У исполнителя существует две команды: **Вперёд n** (где n – целое число), вызывающая передвижение Черепашки на n единиц в том направлении, куда указывает её голова, и **Направо t** (где t – целое число), вызывающая изменение направления движения на t градусов по часовой стрелке.*

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

*Черепашке был дан для исполнения следующий алгоритм:
Повтори 7 [Вперёд 10 Направо 120].*

Определите, сколько точек с целочисленными координатами будут находиться внутри области, ограниченной линией, заданной данным алгоритмом. Точки на линии учитывать не следует.

В демонстрационном варианте КИМ ЕГЭ 2024 года есть изменения, связанные с разрешенными к использованию языками программирования. Из-за исключения из списка школьного языка программирования для решения данного задания возможно использование языка программирования Python, используя модуль Turtle для построения заданной области.

```
from turtle import *
k=20
lt(90)
for i in range(7):
    forward(10*k)
    rt(120)

penup()

for x in range(-5, 15):
    for y in range(-5, 15):
        setpos(x*k, y*k)
        dot(4, 'red')
```

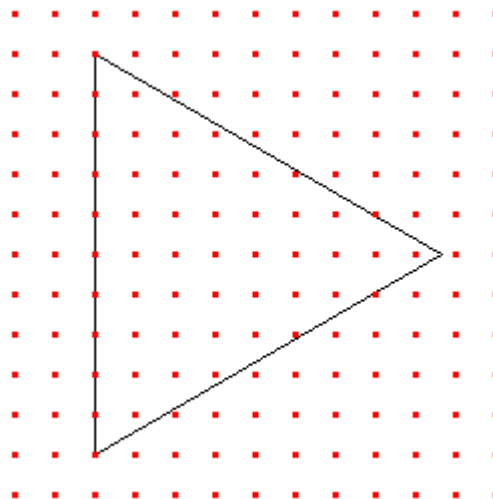
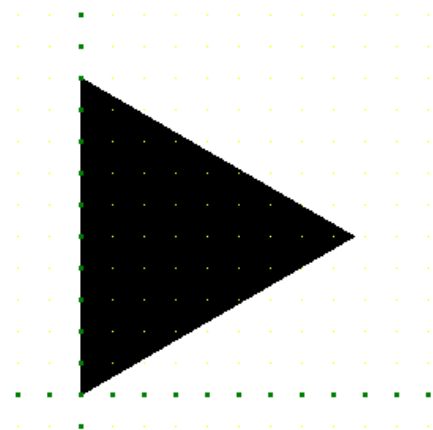


Рисунок 6. Программа на Python для решения задания 6 с ручным подсчетом точек

Если в условии задания необходимо посчитать точки с целочисленными координатами, которые находятся внутри области, но не включая точки, попавшие на границу фигуры, то можно воспользоваться автоматизированным способом подсчета. Данный способ не подходит для подсчета точек на границе фигуры.

```
from turtle import *
tracer(0)
k=20
lt(90)
fillcolor('black')
begin_fill()
for i in range(3):
    forward(10*k)
    rt(120)
end_fill()
up()
g= getcanvas()
count=0
for x in range(-30,30):
    for y in range(-30,30):
        info = g.find_overlapping(x*k, y*k,x*k, y*k)
        if len(info)==1 and info[0]==5:
            count+=1
print(count)
#просто для наглядности, в них необходимости нет
for x in range(-5,15):
    for y in range(-5,15):
        setpos(x*k, y*k)
        if x==0 or y==0:
            dot(4, 'green')
        else:
            dot(2, 'yellow')
```



```
==== RESTART: C:/Users/sdv/AppData/Local/Programs/Python/Python312/
38
```

Рисунок 7. Программа на Python для решения задания 6

Результаты выполнения задания 6 ухудшились. В 2023 году при его выполнении требуется проанализировать результат работы простейшего алгоритма управления исполнителем «Черепашка», а не выполнить простейшую программу, записанную на формальном языке программирования. Видимо, при подготовке к ЕГЭ выпускники

делают упор на программирование, которое помогает в решении более 80% заданий КИМ, а к простейшим исполнителям отношение скорее скептическое. В последствии недостаточно навыков для выполнения этого задания.

Задание 7 базового уровня сложности проверяет умение определять объем памяти, необходимый для хранения графической и звуковой информации, без использования специального ПО за 5 минуты. Задание относится к кодированию информации.

Пример задания.

Музыкальный фрагмент был записан в формате моно, оцифрован и сохранён в виде файла без использования сжатия данных. Размер полученного файла – 28 Мбайт. Затем тот же музыкальный фрагмент был записан повторно в формате стерео (двухканальная запись) и оцифрован с разрешением в 3,5 раза выше и частотой дискретизации в 2 раза меньше, чем в первый раз. Сжатие данных не производилось. Укажите размер полученного при повторной записи файла в Мбайт. В ответе запишите только целое число, единицу измерения писать не нужно.

Задание требует знание теоретического материала:

1) графическая информация

- для хранения растрового изображения нужно выделить в памяти $I = N \cdot i$ битов, где N – количество пикселей и i – глубина цвета (разрядность кодирования);

- количество пикселей изображения N вычисляется как произведение ширины рисунка на высоту (в пикселях);

- глубина кодирования – это количество бит, которые выделяются на хранение цвета одного пикселя;

- при глубине кодирования i битов на пиксель код каждого пикселя выбирается из 2^i возможных вариантов, поэтому можно использовать не более 2^i различных цветов.

2) звуковая информация:

- частота дискретизации определяет количество отсчетов, запоминаемых за 1 секунду; 1 Гц (один герц) – это один отсчет в секунду, а 8 кГц – это 8000 отсчетов в секунду;

- глубина кодирования – это количество бит, которые выделяются на один отсчет;

- для хранения информации о звуке длительностью t секунд, закодированном с частотой дискретизации f Гц и глубиной кодирования V бит требуется $V \cdot t \cdot f$ бит памяти; например, при $f=8$ кГц, глубине кодирования 16 бит на отсчёт и длительности звука 128 секунд требуется;

- при двухканальной записи (стерео) объем памяти, необходимый для хранения данных одного канала, умножается на 2.

3) скорость передачи информации:

- любой канал связи имеет ограниченную пропускную способность (скорость передачи информации), это число ограничивается свойствами аппаратуры и самой линии (кабеля);

- объем переданной информации Q вычисляется по формуле $Q = q \cdot t$, где q – пропускная способность канала (в битах в секунду или подобных единицах), а t – время передачи.

При решении данного задания необходимо обратить внимание на согласованность единиц измерений, например, скорость в битах/с, а размер файла в байтах, а также на размерность получаемой величины (для дополнительной проверки вычислений). Если при решении задачи, приходится работать с достаточно большими числами, желательно выполнить их преобразование через степень двойки, тогда вычисления будут проще: $512\ 000\ \text{бит/с} = 2^9 \cdot 10^3\ \text{бит/с}$.

Задание 8 базового уровня сложности проверяет знание основных понятий и методов, используемых при измерении количества информации без использования специального ПО. Задание относится к кодированию информации. Типы задач разнообразны, но все они сводятся к составлению слов из заданного алфавита по указанной длине и отбору тех, которые подходят нам по определённым условиям.

Пример задания.

Определите количество пятизначных чисел, записанных в восьмеричной системе счисления, в записи которых только одна цифра 6, при этом никакая нечётная цифра не стоит рядом с цифрой 6.

Комбинаторные задачи преимущественно решаются или вручную, анализируя задание, или с помощью написания программы, используя алгоритм составления всех комбинаций слов заданной длины из алфавита:

- 1) завести переменную, которая будет являться хранилищем всех букв алфавита согласно заданию;
- 2) прописать такое количество вложенных циклов for, какая длина слова указана в задании. Счетчик циклов будет идти по символам переменной алфавита;
- 3) в последнем цикле создать переменную, которая будет являться результатом конкатенации всех переменных-счетчиков циклов;
- 4) используя условный оператор, прописать отбор слов, удовлетворяющих условию задачи.

Так же при решении задач на составление слов путем выбора букв из набора или путем перестановки букв в слове можно решить с использованием модуль `itertools` языка Python.

Назначение	Код
Для импорта необходимых функции необходимо их импортировать из модуля	<code>from itertools import product, permutations, combinations</code>
Для получения всевозможных комбинаций определенной длины для одного множества	<code>a = 'AB'</code> <code>print(*product(a, repeat=3))</code> # параметр <code>repeat</code> отвечает за длину слова. Результат работы функции — набор кортежей: (<code>'A', 'A', 'A'</code>) (<code>'A', 'A', 'B'</code>) (<code>'A', 'B', 'A'</code>) (<code>'A', 'B', 'B'</code>) (<code>'B', 'A', 'A'</code>) (<code>'B', 'A', 'B'</code>) (<code>'B', 'B', 'A'</code>) (<code>'B', 'B', 'B'</code>)
	<code>cmb = list(combinations('ABC', 2))</code> <code>print(cmb)</code> Результат работы функции – список: [(<code>'A', 'B'</code>), (<code>'A', 'C'</code>), (<code>'B', 'C'</code>)]
Для получения всевозможных перестановок	<code>print(*permutations('ABC'))</code> Результат работы функции — набор кортежей: (<code>'A', 'B', 'C'</code>) (<code>'A', 'C', 'B'</code>) (<code>'B', 'A', 'C'</code>) (<code>'B', 'C', 'A'</code>) (<code>'C', 'A', 'B'</code>) (<code>'C', 'B', 'A'</code>)

Модуль `itertools` доступен в Python «из коробки», т.е. его нет необходимости устанавливать дополнительно и доступен с версии 3.6.

Задание 9 относится к базам данных, электронным таблицам и текстовым процессорам и проверяет умение обрабатывать числовую информацию в электронных таблицах.

В электронной таблице произвести проверку строк, содержащих числа, удовлетворяющие условию задания и подсчитать количество таких строк. Для решения

этого задания необходимо написать формулы с использованием специальных функций, с помощью которых производятся необходимые вычисления. Хотя это задание отнесено к базовому уровню сложности, но на самом деле его сложность очень высокая, что объясняет наименьшую решаемость (16,92%) задания среди данного типа.

Пример задания.

Откройте файл электронной таблицы, содержащей в каждой строке шесть натуральных чисел. Определите количество строк таблицы, содержащих числа, для которых выполнены оба условия:

– в строке только одно число повторяется ровно два раза, остальные числа различны;

– среднее арифметическое неповторяющихся чисел строки не больше суммы повторяющихся чисел.

	A	B	C	D	E	F
1	37	83	24	19	37	41
2	59	77	43	43	118	38
3	6	65	40	22	6	130
4	63	77	76	8	63	51
5	74	47	97	26	222	23
6	48	24	17	7	24	36
7	27	39	77	35	27	13
8	35	41	6	21	52	41
9	46	32	95	3	23	32
10	34	43	72	42	17	43
11	2	57	76	39	4	57
12	61	90	53	26	61	135
13	89	95	54	7	89	95
14	9	64	76	12	9	21
15	38	60	40	25	12	180
16	89	49	74	2	44	49

В ответе запишите только число.

Для решения данного задания необходимо уметь производить вычисления в электронных таблицах, формулировать логические условия, находить минимум и максимум, сортировать данные по заданным полям, использовать фильтры, а также уметь применять логические, арифметические и статистические функции. В последние годы стало достаточно сложным, для получения правильного ответа необходимо разработать алгоритм (последовательность шагов), выполнить их аккуратно, не допустив при этом неточностей и ошибок.

Типичных ошибок при решении не обнаружено. Либо вычисления неверны, либо экзаменуемые и не брались за решение задачи. Как ни парадоксально, при условии, что это задание базового уровня сложности, направленное на контроль умений выполнять базовые вычислительные операции, применяемые и в других заданиях, процент его выполнения крайне низок во всех группах участников ЕГЭ. Это говорит об одном – при изучении темы «Табличные вычисления» уделяется недостаточно внимания овладению умением составлять математические модели, описывающие условия, заданные для числовых данных. Отсюда и неумение составить нужные формулы для табличных вычислений. Тогда как именно формализация и моделирование два важных начальных этапа при разработке алгоритмов решения задач на ЭВМ.

Самые высокие результаты, как обычно, экзаменуемые показывают при выполнении заданий базового уровня на применение известных алгоритмов в стандартных ситуациях, например, задание 10 – информационный поиск средствами операционной системы или текстового процессора.

Пример задания.

Текст произведения Льва Николаевича Толстого «Севастопольские рассказы» представлен в виде файлов различных форматов. Откройте один из файлов и определите, сколько раз встречается в тексте отдельное слово «теперь» со строчной буквы. Другие формы этого слова учитывать не следует. В ответе запишите только число.

Для осуществления поиска нужной информации в документе используется стандартный поиск, который подразумевает ввод запроса с дальнейшим выделением соответствующих результатов. Дополнительные параметры, такие как форматирование текста и специальные знаки, при этом не учитываются.

Расширенный поиск, в отличие от стандартного, позволяет находить интересующую в документе информацию с учетом множества дополнительных параметров, например, учитывая при проверке регистр символов, подстановочные знаки и даже произношение слов. Важным является тот факт, что в каждом текстовом редакторе расширенный поиск осуществляется в основном одинаково. Прежде всего, необходимо открыть расширенный поиск. Делается это непосредственно в окне «Найти и заменить». Находясь в нужном меню, щелкните по кнопке «Больше», чтобы развернуть весь список параметров.

Так же можно использовать программу на Python, которая считает файл целиком, разобьёт его на слова, переведёт все символы в нижний регистр (к строчным буквам) и считает сколько заданных слов оказалось в этом списке.

```
import re
n = 0
for x in re.split('\W+', open('10-0.txt', 'r').read()).lower():
    if x == 'долг':
        n += 1
print(n)
```

Рисунок 8. Программа на Python для решения задания 10

Задание содержало элемент усложнения (нужно найти количество слов, в которых содержится заданное подслово), согласно которому окончательный ответ мог быть получен не как ответ на запрос поиска, а в результате дополнительного шага с привлечением элементарных сведений из теории множеств. Это не позволило части участников справиться с заданием. Типичными ошибками в этом задании явились: часть участников из полученного количества (слов с заданным подсловом) не вычли количество слов целиком, часть участников наоборот, в качестве ответа дали количество слов целиком.

Задание 19 базового уровня сложности проверяет умение анализировать алгоритм логической игры без использования специального ПО и относиться к тематическому блоку «Основы логики».

Группа заданий (19-21) объединена общей частью условия, в которой сформулированы правила игры, приведённой в задании 19. У этого задания довольно длинное условие. Для их решения внимательно прочитайте его, убедитесь, что вам полностью понятны правила логической игры. Задание 19 из этой группы обычно самое простое, и выполнить его нетрудно; для выполнения задания 20 может потребоваться анализ возможных ходов игроков с помощью дерева игры. Для выполнения задания 21, если его решение не следует из результатов выполнения задания 20, следует построить на черновике полное или неполное дерево игры в виде схемы или таблицы.

Пример задания.

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу один камень или увеличить количество камней в куче в два раза.

Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 129. Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу из 129 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 128$. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.

Отличие задания 2023 года от задания прошлого года заключалось в количестве вариантов ходов, которые могли использовать игроки: их стало 3 вместо 2. Метод решения заданий от этого не меняется, но дерево игры имеет большую ширину и соответственно при ручном решении требуется больший перебор при анализе. При программном решении заданий изменения в количестве вариантов ходов существенной роли не играет. Отмеченные выше изменения в формулировке повлекли снижение результатов в группах слабых участников, однако на более подготовленных это не отразилось.

Задания повышенного уровня сложности. Первое задание повышенного уровня сложности, которое мы видим в КИМ, – это задание 11, которое относится к темам по кодированию информации и проверяет умение подсчитывать информационный объем сообщения. С данным заданием справляется более 50% участников экзамена.

Пример задания. При регистрации в компьютерной системе каждому объекту присваивается идентификатор, состоящий из 250 символов и содержащий только десятичные цифры и символы из 1650-символьного специального алфавита. В базе данных для хранения каждого идентификатора отведено одинаковое и минимально возможное целое число байт. При этом используется посимвольное кодирование идентификаторов, все символы кодируются одинаковым и минимально возможным количеством бит.

Определите объём памяти (в Кбайт), необходимый для хранения 65 536 идентификаторов. В ответе запишите только целое число – количество Кбайт.

Как и задание 7, решение данного задания требует знание теоретического материала (см. задание 7):

- с помощью K бит можно закодировать $Q=2^K$ различных вариантов (чисел);
- таблица степеней двойки, она же показывает, сколько вариантов Q можно закодировать с помощью K бит:

K , бит	1	2	3	4	5	6	7	8	9	10
Q , вариантов	2	4	8	16	32	64	128	256	512	1024

- при измерении количества информации принимается, что в одном байте 8 бит, а в одном килобайте (1 Кбайт) = 1024 байта, в мегабайте (1 Мбайт) = 1024 Кбайта;

- чтобы найти информационный объем сообщения (текста) I , нужно умножить количество символов (отсчетов) N на число бит на символ (отсчет) K : $I=N \cdot K$;

- мощность алфавита M – это количество символов в этом алфавите;

- если алфавит имеет мощность M , то количество всех возможных «слов» (символьных цепочек) длиной N (без учета смысла) равно $Q=M^N$; для двоичного кодирования (мощность алфавита M – 2 символа) получаем известную формулу: $Q=2^N$.

При решении данного задания необходимо обратить внимание на согласованность единиц измерений, а также на размерность получаемой величины (для дополнительной проверки вычислений). Если при решении задачи, приходится работать с достаточно большими числами, желательно выполнить их преобразование через степень двойки, тогда вычисления будут проще: $512\ 000\ бит = 2^9 \cdot 10^3\ бит$.

В задании 12 повышенного уровня сложности необходимо проанализировать алгоритм для формального исполнителя (Редактор) с заданной системой команд. Аналогичное задание присутствует в ЕГЭ уже несколько лет, до компьютерной формы оно выполнялось вручную, теперь появилась возможность выполнения задания программным путем. В 2022 году в задании, как и в прошлые годы, требовалось определить, что получится в результате работы алгоритма для конкретного входа. И с такой постановкой участники относительно неплохо справлялись.

Пример задания.

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

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 (> 1) ИЛИ нашлось (> 2) ИЛИ нашлось (> 0)

ЕСЛИ нашлось (> 1)

ТО заменить (> 1, 22>)

КОНЕЦ ЕСЛИ

ЕСЛИ нашлось (> 2)

ТО заменить (> 2, 2>)

КОНЕЦ ЕСЛИ

ЕСЛИ нашлось (> 0)

ТО заменить (> 0, 1>)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

На вход приведённой выше программе поступает строка, начинающаяся с символа «>», а затем содержащая 39 цифр «0», n цифр «1» и 39 цифр «2», расположенных в произвольном порядке. Определите наименьшее значение n , при котором сумма числовых значений цифр строки, получившейся в результате выполнения программы, является простым числом.

В 2023 году в задание был добавлен элемент анализа, вместо нахождения результата работы алгоритма в задании требуется подобрать значение входного параметра, чтобы результат алгоритма удовлетворял заданным требованиям. Задание в аналогичной формулировке было представлено в демоверсии.

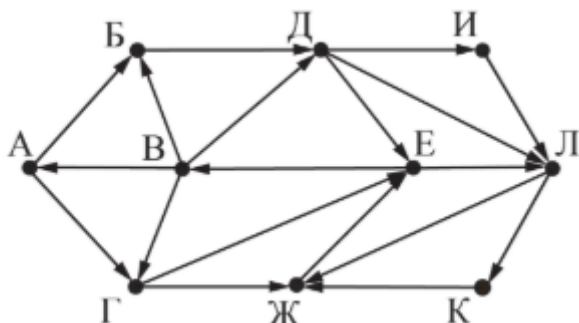
Возможность программного выполнения при этом сохраняется, но требует грамотного добавления перебора. Поскольку при сдаче ЕГЭ в компьютерной форме

доступны среды программирования, проще всего написать программу, которая моделирует исполнителя Редактор; для этой цели лучше всего подходит язык Python, обладающий широким набором встроенных функций для обработки символьных строк. Дополнительный усложняющий элемент содержания явился причиной значительного падения результативности по заданию.

Задание 13 повышенного уровня сложности проверяет умение представлять и считывать данные в разных типах информационных моделей. Отличительной особенностью задания 13 являлось то, что требовалось найти наибольшую длину пути из начального пункта в конечный, тогда как в предыдущие годы задача состояла в подсчете количества дорог. Это изменение не повлекло падение результата выполнения задания. Кроме того, в 2022 году в задании начальный пункт совпадал с конечным, в 2023 году оба эти пункта различны.

Пример задания.

На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.



Определите количество различных путей ненулевой длины, которые начинаются и заканчиваются в городе Е, не содержат этот город в качестве промежуточного пункта и проходят через промежуточные города не более одного раза.

Несмотря на то, что в задании 13 (открытый вариант 2023 года) была изменена формулировка: вместо подсчета количества путей необходимо было найти длину самого длинного пути, участники справились с этим заданием. Метод решения задания в измененной формулировке практически такой же с небольшими модификациями. Среди участников, выполнявших задание 13 из открытого варианта, чуть более 60% выполнили задание верно, остальные получили неверный ответ, потому что фактически выполняли другую задачу, и вместо нахождения длины самого длинного пути находили количество дорог. Внимательное чтение задания является первым шагом на пути к успеху, что, к сожалению, не учли эти ребята.

В демонстрационном варианте КИМ ЕГЭ 2024 года есть изменения, связанные с заданием 13: будет проверяться умение использовать маску подсети при адресации в соответствии с протоколом IP.

Пример задания (демонстрационный вариант КИМ ЕГЭ по информатике 2024 год).

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и маске сети. Сеть задана IP-адресом 192.168.32.160 и маской сети 255.255.255.240. Сколько в этой сети IP-адресов, для которых сумма единиц в двоичной записи IP-адреса чётна? В ответе укажите только число.

Для решения данного типа заданий необходимо понимать теоретические основы IP-адресации. Согласно ним, для расчета адреса подсети по известному IP адресу одного из устройств и маске подсети необходимо:

- преобразовать IP-адрес и маску сети в двоичную систему счисления: как правило, IP-адрес и маска сети представлены в десятичной системе счисления.

- применить маску сети к IP-адресу: выполнить логическое «И» (AND) между двоичными представлениями IP-адреса и маски сети. Это означает, что для каждой пары битов в IP-адресе и маске сети, результирующий бит будет равен 1, только если оба исходных бита равны 1. Это позволит определить адрес подсети.

- преобразовать результат обратно в десятичную систему: после выполнения операции «И» полученное двоичное значение переводят обратно в десятичную систему счисления. Это будет адресом подсети.

Как и многие другие задания ЕГЭ по информатике возможно выполнить «вручную» или с помощью программных средств. Для решения задания можно воспользоваться написанием программы на Python с использованием модуля `ipaddress`. Модуль предоставляет удобные инструменты для работы с IP-адресами и сетями, включая вычисление адреса подсети. А в саму программу так же можно включить перебор всех адресов для соответствия результатов условию задания.

Задание 14 повышенного уровня сложности проверяет знание позиционных систем счисления без использования специального ПО в рамках темы «Системы счисления».

Среди участников, выполнявших это задание из открытого варианта, только 47% выполнили его верно. Использование двузначного основания системы счисления не позволило части ребят справиться с данным заданием, хотя задание похоже на задание прошлого года, и на представленное в демоверсии.

Пример задания.

Операнды арифметического выражения записаны в системе счисления с основанием 15.

$$123 \times 5_{15} + 1 \times 233_{15}$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 15-ричной системы счисления. Определите наименьшее значение x , при котором значение данного арифметического выражения кратно 14. Для найденного значения x вычислите частное от деления значения арифметического выражения на 14 и укажите его в ответе в десятичной системе счисления. Основание системы счисления в ответе указывать не нужно.

Выполнение данного задания требует знания:

- позиционных систем счисления, алфавита системы счисления, основания системы счисления;

- алгоритма перевода числа из любой системы счисления в десятичную и наоборот (желательно понимать, как данные алгоритмы реализуются или с помощью электронных таблиц, или с помощью написания программы);

- действий в позиционных системах счисления с одинаковым основанием (сложение, вычитание, умножение, деление).

Как и многие другие задания в экзамене, данное задание возможно решать с помощью аналитического способа, методом перебора, реализованного с использованием электронных таблиц или написанием программы.

Задание 15 в 2023 году имеет наименьшую решаемость в блоке заданий, посвящённых математической логике и логическому анализу игры, но сам процент выполнения, по сравнению с 2022 годом, вырос.

Пример задания.

Обозначим через $\text{ДЕЛ}(n, t)$ утверждение «натуральное число n делится без остатка на натуральное число t ». Для какого наименьшего натурального числа A формула

$$(\text{ДЕЛ}(x, 2) \rightarrow \neg \text{ДЕЛ}(x, 3)) \vee (x + A \geq 100)$$

тождественно истинна (т.е. принимает значение 1) при любом натуральном значении переменной x ?

Для решения заданий 15 необходимо знать элементы математической логики: преобразование таблиц истинности, законы алгебры, логики, порядок действий.

Для решения заданий с отрезками $\text{ДЕЛ}()$ необходимо знать тему о множествах. Для упрощения решений можно пользоваться следующими законами, описанными в статье Константина Юрьевича Полякова (<https://kpolyakov.spb.ru/download/del18.pdf>).

Задание 16 - это задание, которое требует вычисления рекуррентных выражений с использованием среды программирования и входит в список заданий по теме алгоритмизация и программирование.

Пример задания.

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(n) = 1 \text{ при } n = 1;$$

$$F(n) = n \times F(n - 1), \text{ если } n > 1.$$

Чему равно значение выражения $F(2023) / F(2020)$?

Задачи, в которых требуется найти значение заданной рекурсивной функции при известных значениях параметров можно решать с помощью ручных вычислений, используя электронные таблицы или с помощью своей программы; последние два способа обычно более эффективны. В этом случае надо просто переписать алгоритм с использованием синтаксиса выбранного языка программирования. Однако для данного задания такой способ не подходит, поскольку в нём используются большие числовые значения, и рекурсия будет работать очень долго, и при этом исчерпает имеющуюся память.

Само задание достаточно простое, в нём просто подсчитывается сумма арифметической прогрессии. Аналитический способ или программирование с запоминанием вычисленных значений без их перевычислений приведет к правильному ответу. Следует обращать внимание учащихся на то, что далеко не всегда то, что кратко записывается, быстро работает. И что рекурсию надо применять очень осторожно, тщательно взвесив все «за» и «против».

Задание 17 повышенного уровня сложности, которое тематически относится к алгоритмизации и программированию, и проверяет умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования.

Пример задания.

В файле содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Определите количество пар последовательности, в которых только одно число оканчивается на 3, а сумма квадратов элементов пары не меньше квадрата максимального элемента последовательности, оканчивающегося на 3. В ответе запишите два числа: сначала количество найденных пар, затем максимальную из сумм квадратов элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Для решения данного задания необходимо умение написания базовых алгоритмов (проверка на простоту, нахождение делителей, поиск минимального/максимального, нахождение суммы/произведения чисел, разбиение числа на цифры, нахождение

среднего арифметического, алгоритмы сортировки и т.д.) с использованием выбранного языка программирования.

Для успешного выполнения этого задания необходимо свободно владеть базовыми навыками программирования, в том числе чтением данных из файлов и обработкой массивов.

Часто к ошибкам в таких заданиях приводит невнимательное прочтение условия: экзаменуемые находят минимум, вместо максимума и наоборот, сумму чисел, вместо количества, выполняют операцию меньше или строго больше, вместо «не меньше» и пр.

Задание 18 проверяет умение использовать электронные таблицы для обработки целочисленных данных. Отличительной особенностью задания повышенного уровня сложности, в котором требуется проанализировать алгоритм для Исполнителя Робот, являлось то, что поле Робота содержало несколько конечных клеток (конечными клетками являются те, из которых Робот более не может продолжать движение). Этот факт является элементом усложнения задания, ребята справились с этим заданием хуже, чем в прошлые годы.

Пример задания.

Квадрат разлинован на $N \times N$ клеток ($1 < N < 30$). Исполнитель Робот может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: вправо или вниз. По команде вправо Робот перемещается в соседнюю правую клетку, по команде вниз – в соседнюю нижнюю. Квадрат ограничен внешними стенами. Между соседними клетками квадрата также могут быть внутренние стены. Сквозь стену Робот пройти не может. Перед каждым запуском Робота в каждой клетке квадрата лежит монета достоинством от 1 до 100. Посетив клетку, Робот забирает монету с собой; это также относится к начальной и конечной клеткам маршрута Робота. Определите максимальную и минимальную денежные суммы, которые может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.

Исходные данные представляют собой электронную таблицу размером $N \times N$, каждая ячейка которой соответствует клетке квадрата. Внутренние и внешние стены обозначены утолщенными линиями.

Пример входных данных:

1	8	8	4
1	1	1	3
0			
1	3	1	2
		2	
2	3	5	6

В этом задании, нужно найти оптимальный путь для Робота, который перемещается на клетчатом поле. Робот может на каждом шаге выбирать одно из нескольких направлений движения (например, только вправо и вниз).

В каждой клетке Робот получает некоторую награду (например, «берёт монету»), и нужно найти такой путь, при котором общая награда будет наибольшая (или наименьшая, если это не награда, а штраф).

Теоретически можно решить такую задачу полным перебором вариантов: рассмотреть все возможные пути и выбрать лучший. Однако количество возможных путей для полей даже не очень большого размера слишком велико для того, чтобы решить эту задачу за время проведения ЕГЭ, даже если удастся безошибочно написать программу для такого перебора.

Данная задача успешно и быстро решается с помощью динамического программирования – метода оптимизации, который предложил американский математик Ричард Беллман. Он сформулировал очень простой принцип оптимальности пути: любая часть оптимального пути оптимальна.

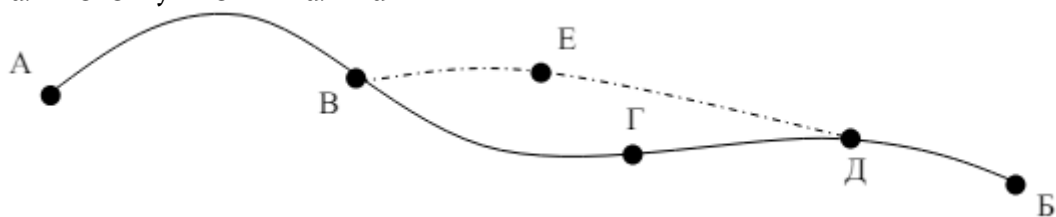


Рисунок 9. Принцип оптимальности пути

Например, пусть мы нашли оптимальный путь из точки А в точку Б, который проходит через точки В, Г и Д: Принцип Беллмана утверждает, что, например, путь ВГД – это оптимальный путь из В в Д. Если бы это было не так и существовал бы другой, лучший путь между В и Д (например, ВЕД на рисунке), то и путь АВГДБ не был бы оптимальным.

Решение задачи сводится к построению вспомогательной таблицы, где для поиска максимального значения основной формулой является формула вида $=\text{МАКС}(A23;B22)+A1$, где выбирается максимальная из накопленных слева или сверху от расчетной ячейки сумм (B22;A22) и складывается со значением в текущей ячейке (соответствующей расчетной) основной таблицы (A1).

	A	B	C
1	27	29	25
2	27	29	25
3	30	30	26
4	25	30	27
5	27	30	27
6	28	30	28
7	29	25	25
8	26	28	28
9	29	26	30
10	26	28	28
11	30	30	27
12	27	26	29
13	25	29	27
14	27	30	30
15	26	27	28
16	27	27	26
17	29	30	30
18	29	30	26
19	30	30	28
20	25	27	30
21			
22			
23		$=\text{МАКС}(A1;B22)$	56
24		54	85
25		84	115
26		109	145
27		136	175
28		164	205
29		193	230
30		219	258

00	01	107	134	164	190	217	248
85	106	134	164	190	217	248	
115	132	158	188	216	243	268	
145	159	188	215	242	272	300	
175	186	216	$=F26+E5$	269	298	328	
205	214	241	268	294	327	353	
230	239	269	298	324	354	380	
258	267	298	324	354	383	412	
284	314	343	351	382	411	437	
312	342	368	377	409	441	468	
342	369	398	403	435	471	496	
368	398	423	432	460	501	528	
397	425	452	461	486	531	558	

56	81	107	134	162	192	222
85	106	134	164	190	217	248
115	132	$=D25+D3$	188	216	243	268
145	159	188	215	242	272	300
175	186	216	243	269	298	328
205	214	241	268	294	327	353
230	239	269	298	324	354	380
258	267	298	324	354	383	412
284	314	343	351	382	411	437
312	342	368	377	409	441	468
342	369	398	403	435	471	496

Рисунок 10. Заполнение расчетной таблицы

Затем в ячейках справа от стен делаются поправки: выбор МАКС заменяется на сумму сверху от расчетной ячейки, а в ячейках снизу от стен делаются поправки: выбор МАКС заменяется на сумму слева от расчетной ячейки.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	27	29	25	26	27	28	30	30	27	26	27	30	28	25	29	30	26	25	27	28
2	27	29	25	27	30	26	25	26	25	30	25	25	30	25	30	28	28	29	27	27
3	30	30	26	26	30	28	27	25	29	30	26	27	27	26	30	30	25	26	29	29
4	25	30	27	29	27	26	29	28	29	25	27	29	29	29	27	29	29	27	30	26
5	27	30	27	28	28	26	26	28	27	25	28	30	28	27	28	28	28	29	30	28
6	28	30	28	25	25	25	29	25	30	25	27	29	27	26	30	29	25	29	27	28
7	29	25	25	28	30	26	27	26	30	26	27	25	25	29	26	29	29	28	30	30
8	26	28	28	29	26	30	29	29	25	29	26	26	30	25	29	28	25	27	30	28
9	29	26	30	29	27	28	28	25	28	25	25	27	29	29	27	26	29	27	28	26
10	26	28	28	25	26	27	30	27	27	27	30	26	30	25	25	26	29	25	26	28
11	30	30	27	29	26	26	30	25	25	30	26	27	27	25	25	25	30	28	26	25
12	27	26	29	25	29	25	30	27	25	29	29	26	25	30	29	25	28	30	26	26
13	25	29	27	27	29	25	30	27	25	29	25	26	30	27	26	27	30	26	27	28
14	27	30	30	26	28	27	25	29	26	27	30	26	28	28	28	30	26	29	27	30
15	26	27	28	27	26	28	26	26	26	28	29	27	26	28	29	30	30	29	27	29
16	27	27	26	30	26	29	27	28	30	27	28	30	26	29	29	28	28	26	26	25
17	29	30	30	29	30	26	28	26	28	30	25	29	25	27	26	25	25	29	27	28
18	29	30	26	28	26	26	29	28	29	29	25	27	25	29	27	29	25	29	29	28
19	30	30	28	27	30	27	28	26	27	25	26	25	28	26	28	26	27	27	27	25
20	25	27	30	26	29	28	25	29	27	27	27	27	27	29	25	25	30	26	29	25

27	56	81	107	134	162	192	222	249	275	302	332	360	385	414	444	470	495	522	550
54	85	106	134	164	190	217	248	274	305	330	357	390	415	445	473	501	530	557	584
84	115	132	158	188	216	243	268	303	335	361	388	417	443	475	505	530	556	586	615
109	145	159	188	215	242	272	300	332	360	388	417	446	475	502	534	563	590	620	646
136	175	186	216	243	269	298	328	359	385	416	447	475	502	530	562	591	620	650	678
164	205	214	241	268	294	327	353	389	410	443	476	502	528	560	591	616	649	677	706
193	230	239	269	298	324	354	380	419	436	470	501	527	557	586	620	649	677	707	737
219	258	267	298	324	354	383	412	444	465	496	527	557	582	615	648	674	704	737	765
248	284	314	343	351	382	411	437	472	490	521	554	586	615	642	674	703	731	765	791
274	312	342	368	377	409	441	468	499	526	556	582	616	641	667	700	732	757	791	819
304	342	369	398	403	435	471	496	524	556	582	609	643	668	693	725	762	790	817	844
331	368	398	423	432	460	501	528	553	585	614	640	668	698	727	750	790	820	846	872
356	397	425	452	461	486	531	558	583	614	639	666	698	725	753	777	820	846	873	901
383	427	457	483	511	538	556	587	613	641	671	697	726	754	782	807	846	875	902	932
409	454	485	512	538	566	582	613	639	669	700	727	753	782	811	837	876	905	932	961
436	481	511	542	568	597	609	641	671	698	728	758	784	813	842	865	904	931	958	986
465	511	541	571	601	627	637	667	699	729	754	787	812	840	868	890	929	960	987	1015
494	541	567	599	627	653	666	695	724	753	778	805	830	859	895	919	954	989	1018	1046
524	571	599	626	657	684	712	738	765	790	816	841	869	895	923	949	981	1016	1045	1071
549	598	629	655	686	714	739	768	795	822	849	876	903	932	957	982	1012	1042	1074	1099

Рисунок 11. Результаты вычислений

Для поиска минимального значения достаточно во всех расчетных ячейках заменить МАКС на МИН.

Для успешного выполнения этого задания необходимо понимать суть динамических меняющихся расчетов и свободно владеть базовыми навыками применения встроенных функций табличного процессора и составления формул.

Характерно то, что в 2022 году с этим заданием участники ЕГЭ справились гораздо лучше, чем в 2023 году, связано это с появлением нескольких конечных клеток. Типичная ошибка при выполнении задания 18 связана как раз с этим новшеством: участники в качестве ответа представили количество монет для конечной правой нижней клетки (для формулировки прошлого года), что явилось неверным ответом. Если сравнить с процентом участников, выполнивших задание верно (24%), то становится ясно насколько важно внимательно читать задание и быть уверенным, что решаешь задачу именно в той постановке, которая требуется.

Задание 20 проверяет умение найти выигрышную стратегию и относиться к математической логике и логическому анализу игры.

Пример задания.

Для игры, описанной в задании 19, найдите два наименьших значения S , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Найденные значения запишите в ответе в порядке возрастания

Группа заданий (19-21) объединена общей частью условия, в которой сформулированы правила игры, приведённой в задании 19. У этого задания довольно длинное условие. Для их решения необходимо внимательно прочитать условия, убедиться, что полностью понятны правила логической игры. Для выполнения задания 20 может потребоваться анализ возможных ходов игроков с помощью дерева игры.

Задачи на теорию игр (20 и 21) являются одними из наиболее сложных на ЕГЭ по информатике. Несмотря на возможность их решения при помощи компьютерного перебора, изначально разработчики демо-варианта предлагали ручное (аналитическое) решение.

В 2023 году в ЕГЭ по информатике появилось новое задание, которое «призвано привлечь внимание к параллельному программированию, технологиям организации многозадачных / многопоточных вычислений». Для выполнения задания необходимо использовать файл – электронную таблицу. Это задание 22, которое проверяет навык построения математических моделей для решения практических задач.

Пример задания.

В файле содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс B зависит от процесса A , если для выполнения процесса B необходимы результаты выполнения процесса A . В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Типовой пример организации данных в файле:

<i>ID процесса B</i>	<i>Время выполнения процесса B (мс)</i>	<i>ID процесса(-ов) A</i>
<i>1</i>	<i>4</i>	<i>0</i>
<i>2</i>	<i>3</i>	<i>0</i>
<i>3</i>	<i>1</i>	<i>1;2</i>
<i>4</i>	<i>7</i>	<i>3</i>

Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

Типовой пример имеет иллюстративный характер. Для выполнения задания используйте данные из прилагаемого файла.

На самом деле все современные вычисления многопоточные и все современные компьютеры многопроцессорные. Если раньше многопроцессорные вычислительные системы (МВС) применялись в основном в научной сфере для решения вычислительных задач, требующих мощных вычислительных ресурсов, то сейчас МВС применяются повсеместно.

Для понимания данного задания возможно использование диаграммы Ганта (это инструмент управления проектами, иллюстрирующий то, как выполняется запланированная работа с течением времени. Диаграмма Ганта может включать даты начала и завершения заданий, контрольные точки, зависимости между заданиями, исполнителей и не только).

<i>ID процесса B</i>	<i>Время выполнения процесса B (мс)</i>	<i>ID процесса (ов) A</i>
<i>1</i>	<i>4</i>	<i>0</i>
<i>2</i>	<i>3</i>	<i>0</i>
<i>3</i>	<i>1</i>	<i>1; 2</i>
<i>4</i>	<i>7</i>	<i>3</i>
<i>5</i>	<i>6</i>	<i>3</i>
<i>6</i>	<i>3</i>	<i>5</i>
<i>7</i>	<i>1</i>	<i>4; 6</i>
<i>8</i>	<i>2</i>	<i>7</i>

9	7	0
10	8	0
11	6	9
12	6	10

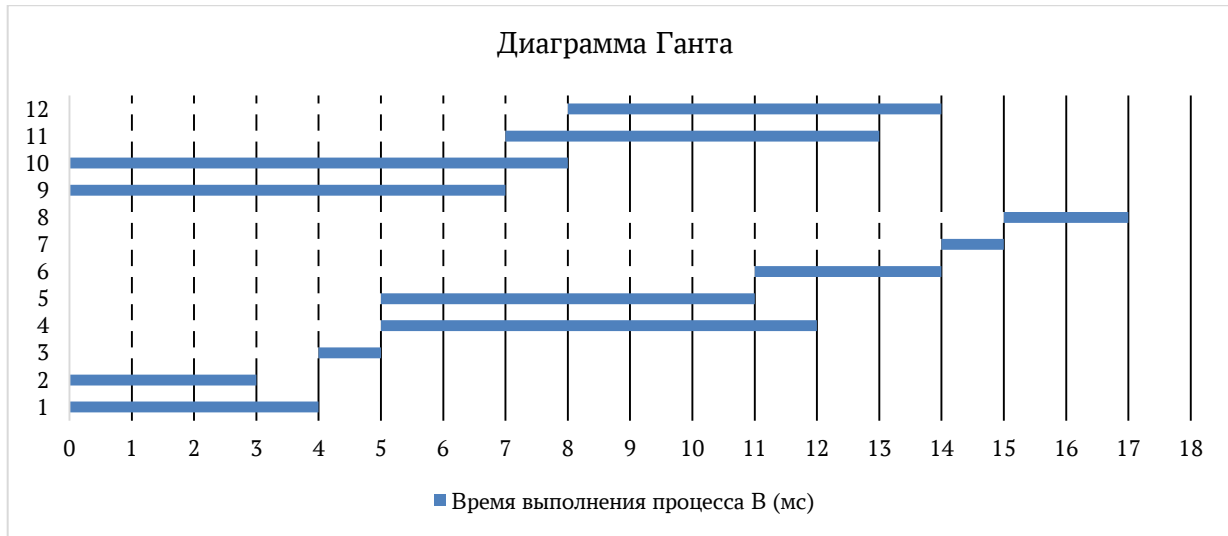


Рисунок 12. Данные и диаграмма Ганта для задания 22

Хотя в данном примере мы можем по построенной диаграмме определить минимальное время выполнения всей совокупности процессов (17 мс), но само построение диаграммы требует достаточного времени и она не применима для больших данных.

Поэтому применимы два других метода для решения заданий. Способ первый - использование графов. Граф, вершинами которого будут ID процессов из первого столбца «ID процесса В», а рёбрами - связи процессов, заданные столбцами «ID процесса(ов) А» и «ID процесса В». Поскольку для выполнения процесса В иногда необходимы результаты выполнения процесса А, рёбра графа будут дугами (в теории графов ребро, имеющее направление, называется ориентированным ребром или дугой), а сам граф - ориентированным графом или орграфом. В столбце «ID процесса(ов) А» содержатся номера вершин графа, являющихся началом дуги, в столбце «ID процесса В» задан конец для каждой дуги (рисунок 13).

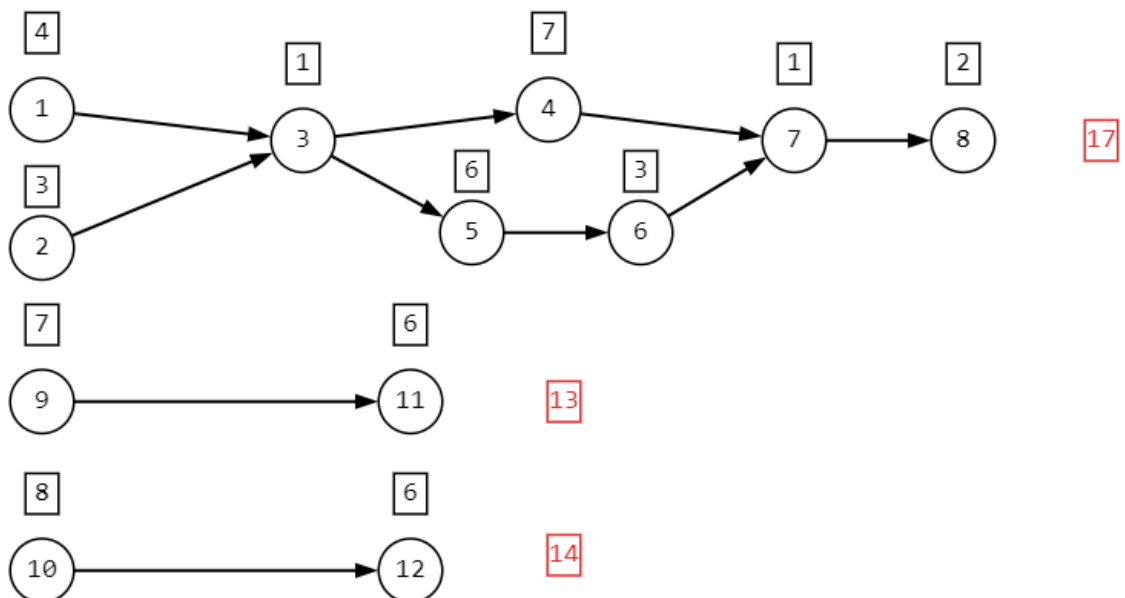


Рисунок 13. Использование графов для решения задачи

Обозначим рядом с каждой вершиной графа время выполнения процесса из второго столбца таблицы «Время выполнения процесса В (мс)» – вес вершины. Получим граф со взвешенными вершинами.

Обратите внимание, что граф для решения задачи состоит из трёх изолированных подграфов, и для корректного решения задачи наибольшую сумму весов следует найти для каждого из изолированных подграфов, после чего следует выбрать наибольшую из найденных сумм, поскольку все процессы должны быть выполнены по условию задачи.

Способ второй – работа с электронной таблицей. Для решения с помощью электронных таблиц можно использовать данный алгоритм, описанный в статье Попова Владислава Сергеевича

(https://www.researchgate.net/publication/368390475_Novaa_zadaca_EGE_po_informatike_No_22_resenie_v_elektronnyh_tablicah_New_USE_task_in_computer_science_No_22_solution_in_spreadsheets).

В среднем процент выполнения задания 22 в Томской составляет 61,43 %, что является достаточно высоким показателем для повышенного уровня сложности задач.

Задание 23 повышенного уровня сложности проверяет умение анализировать результат исполнения алгоритма, содержащего ветвление и цикл без использования специального ПО.

Пример задания.

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

А. Прибавить 1

В. Умножить на 2

Программа для исполнителя – это последовательность команд. Сколько существует программ, для которых при исходном числе 1 результатом является число 35, при этом траектория вычислений содержит число 10 и не содержит 17?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы АВА при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

Есть 4 типа задач согласно каталогу ФИПИ и КИМ прошедшего экзамена:

- 1) Количество программ по заданному числу.
- 2) Количество программ с обязательным этапом.
- 3) Количество программ с избегаемым этапом.
- 4) Количество программ с обязательным и избегаемым этапом.

Основные способы решения задания: аналитический (с помощью дерева вариантов или таблицы) или программный (рекурсия или динамическое программирование).

Задания высокого уровня сложности. Первое задание высокого уровня сложности, которое встречаются обучающиеся, – это задание 21, которое проверяет умение строить дерево игры по заданному алгоритму и находить выигрышную стратегию.

Пример задания (описание игры представлено в задании 19).

Для игры, описанной в задании 19, найдите минимальное значение S , при котором одновременно выполняются два условия:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Если найдено несколько значений S , в ответе запишите минимальное из них.

Задание 24 относится к тематическому блоку Алгоритмизация и программирование и проверяет умение создавать собственные программы для обработки символьной информации. Задание выполняется с использованием прилагаемого файла.

Пример задания.

Текстовый файл состоит из символов A, C, D, F и O.

*Определите максимальное количество идущих подряд пар символов вида
согласная + гласная*

в прилагаемом файле.

Для выполнения этого задания следует написать программу.

Для успешного выполнения этого задания необходимо свободно владеть базовыми навыками программирования, работы с файлами данных, обработки строк, а также способами реализации чтения строки из файла.

Для Python удобнее всего использовать такую конструкцию:

```
with open("k7.txt", "r") as F:  
s = F.readline()
```

конструкция with-as – это контекстный менеджер, в данном случае он открывает указанный файл в режиме чтения (второй аргумент «r» при вызове функции open), записывает ссылку на него в файловую переменную F, выполняет тело блока (читает первую строку файла в переменную s) и закрывает (освобождает) файл.

В языке PascalABC.NET можно выполнить перенаправление потока ввода:

```
assign( input, 'k7.txt' );  
readln(s);
```

программа будет «думать», что читает данные, введённые с клавиатуры (с консоли), а на самом деле эти данные будут прочитаны из файла k7.txt.

В языке FreePascal также можно выполнить перенаправление потока ввода, но нужно дополнительно открывать входной поток:

```
assign( input, 'k7.txt' );  
reset( input );  
readln(s);
```

При работе в среде FreePascal нужно убедиться, что в параметрах компилятора включена поддержка длинных символьных строк; на всякий случай стоит добавить в первой строке программы директиву {\$H+}.

Среда PascalABC не поддерживает работу с длинными символьными строками, поэтому для решения задачи использовать версию PascalABC.NET.

В языке C++ используем потоки:

```
#include  
#include using namespace std;  
int main()  
{  
ifstream F(«k7.txt»);  
string s;  
getline( F, s );  
...  
}
```

Для выполнения заданий требуется уметь определять длину строки, обращаться к символу по индексу, копировать и вставлять строки и подстроки.

```

f = open('ege-inf-demo2023-z24.txt', 'r')
s = f.read()
m = 0
k = 0
i = 0
while i < len(s):
    if s[i] in 'CDF' and s[i+1] in 'AO':
        k += 1
        i += 2
        if k > m:
            m = k
    else:
        k = 0
        i += 1
print(m)

```

Рисунок 14. Пример решения задания с помощью Python

Сложность выполнения этого задания состоит в нахождении правильного алгоритма, основанного на подсчете длин подстрок согласно условию задачи, это одна из причин падения процента выполнения.

Здание 25, в свою очередь, проверяет умение написания собственной программы для обработки целочисленной информации.

Пример задания.

Назовём маской числа последовательность цифр, в которой также могут встречаться следующие символы:

– символ «?» означает ровно одну произвольную цифру;

– символ «» означает любую последовательность цифр произвольной длины; в том числе «*» может задавать и пустую последовательность.*

*Например, маске 123*4?5 соответствуют числа 123405 и 12300405.*

*Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске 1?2139*4, делящиеся на 2023 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 2023. Количество строк в таблице для ответа избыточно.*

Данное задание может решаться и без написания программы, с использованием электронных таблиц. Для этого необходимо лишь составить ряд чисел, соответствующий условию, и воспользоваться фильтром по маске для поиска результата.

Для генерации чисел используется «Прогрессия» (арифметическая прогрессия с начальным значением 2023, шагом 2023 и предельным значением 10000000000).

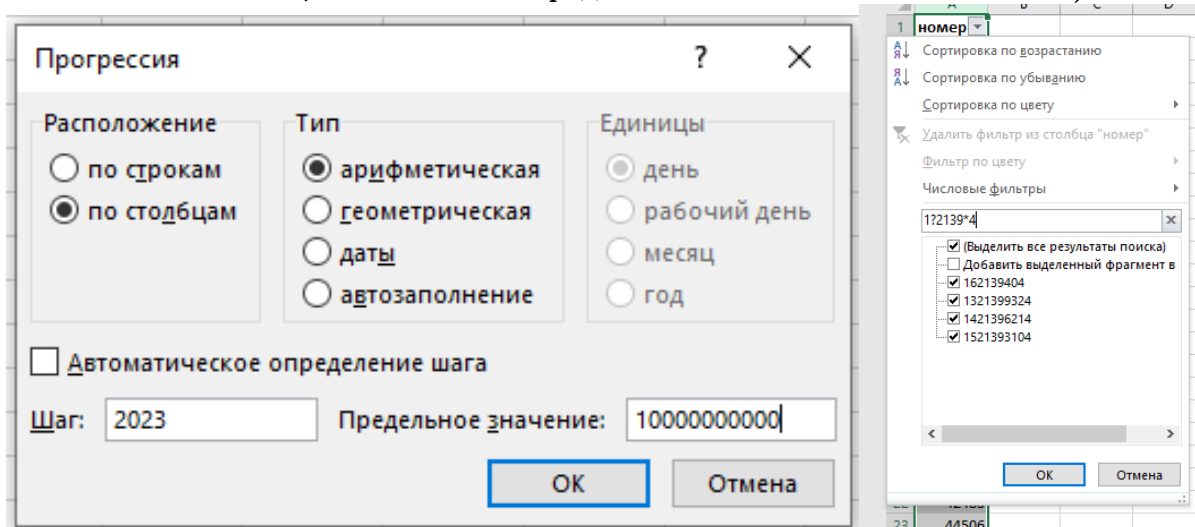


Рисунок 15. Использование табличного процессора для решения задач

Использование электронной таблицы в случае с нахождением делителей тоже возможно, помня, что наименьший из пары делителей, таких что $a * b = n$, не превышает квадратного корня из n , где n – предельное значение. К сожалению, использование таблиц в некоторых типах трудозатратно, и проще написать программу.

Для написания программы и решения задачи необходимо уметь реализовывать на выбранном языке программирования перебор чисел и подсчет соответствий заданному условию (например, делимость чисел, количество делителей, нахождение простых чисел).

Задание 26 участники выполнили намного хуже, чем в прошлом году: только 6% всех участников справились с ним.

Пример задания.

В магазине для упаковки подарков есть N кубических коробок. 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 3 единицы меньше длины стороны другой коробки. Определите наибольшее количество коробок, которое можно использовать для упаковки одного подарка, и максимально возможную длину стороны самой маленькой коробки, где будет находиться подарок. Размер подарка позволяет поместить его в самую маленькую коробку.

Входные данные

В первой строке входного файла находится число N – количество коробок в магазине (натуральное число, не превышающее 10 000). В следующих N строках находятся значения длин сторон коробок (все числа натуральные, не превышающие 10 000), каждое – в отдельной строке.

Запишите в ответе два целых числа: сначала наибольшее количество коробок, которое можно использовать для упаковки одного подарка, затем максимально возможную длину стороны самой маленькой коробки в таком наборе.

Типовой пример организации данных во входном файле

5
43
40
32
40
30

Пример входного файла приведён для пяти 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 30, 40 и 43 или 32, 40 и 43 соответственно, т.е. количество коробок равно 3, а длина стороны самой маленькой коробки равна 32.

Задание выполняется с использованием прилагаемых файлов.

Анализируя условие задания 26 из открытого варианта можно сказать, что задание оказалось сложнее тех, которые использовались в прошлые годы и задания из демоверсии. Задание на экзамене оказалось олимпиадного уровня (по уровню сложности данное задание приближается к заданию 27, которое традиционно считалось самым сложным). Выполнить задание можно с помощью электронной таблицы или программированием. Главное здесь – это алгоритм, инструмент в принципе не важен.

Трудность при выполнении данного задания заключается не только в том, что непросто придумать алгоритм для его решения, а в том, что необходимо построить доказательство корректности алгоритма, убедиться в его правильности, а это является важным этапом.

Не имея достаточных знаний об известных алгоритмах, участник вынужден действовать интуитивно, а в таких заданиях, как представленное в варианте, интуиции доверять достаточно сложно. Доказательство корректности при помощи прогона на тестах также представляется непростой задачей для выполнения на экзамене.

Для успешного выполнения этого задания с помощью программирования необходимо свободно владеть базовыми навыками программирования и работы с файлами данных, навыками обработки числовых последовательностей.

Для выполнения задания средствами электронных таблиц, достаточно уметь сортировать данные и составлять несложные формулы с простейшими логическими и статистическими функциями. Сложность выполнения этого задания состоит в построении правильного «жадного» алгоритма обработки отсортированных значений.

В отличие от предыдущего задания, задание 27 можно выполнять очевидным переборным способом на файле небольшой размерности, и в этом случае наличие такой программы может служить своего рода проверочной программой. Если оба алгоритма (переборный и эффективный) на файле А дают один и тот же ответ, есть надежда, что разработанный эффективный алгоритм верный.

Пример задания.

Все пункты расположены вдоль автомагистрали и имеют номера, соответствующие расстоянию от нулевой отметки до конкретного пункта. Известно количество пробирок, которое ежедневно принимают в каждом из пунктов. Пробирки перевозят в специальных транспортировочных контейнерах вместимостью не более 36 штук. Каждый транспортировочный контейнер упаковывается в пункте приёма и вскрывается только в лаборатории.

Стоимость перевозки биоматериалов равна произведению расстояния от пункта до лаборатории на количество контейнеров с пробирками. Общая стоимость перевозки за день равна сумме стоимостей перевозок из каждого пункта в лабораторию. Лабораторию расположили в одном из пунктов приёма биоматериалов таким образом, что общая стоимость доставки биоматериалов из всех пунктов минимальна. Определите минимальную общую стоимость доставки биоматериалов из всех пунктов приёма в лабораторию.

Входные данные

Дано два входных файла (файл А и файл В), каждый из которых в первой строке содержит число N ($1 \leq N \leq 10\,000\,000$) – количество пунктов приёма биоматериалов. В каждой из следующих N строк находится два числа: номер пункта и количество пробирок в этом пункте (все числа натуральные, количество пробирок в каждом пункте не превышает 1000). Пункты перечислены в порядке их расположения вдоль дороги, начиная от нулевой отметки.

В ответе укажите два числа: сначала значение искомой величины для файла А, затем – для файла В.

Типовой пример организации данных во входном файле

6

1 100

2 200

5 4

7 3

8 2

При таких исходных данных и вместимости транспортировочного контейнера, составляющей 96 пробирок, компании выгодно открыть лабораторию в пункте 2. В этом случае сумма транспортных затрат составит: $1 * 2 + 3 * 1 + 5 * 1 + 6 * 1 + 8 * 2$.

Типовой пример имеет иллюстративный характер. Для выполнения задания используйте данные из прилагаемых файлов. Предупреждение: для обработки файла В не следует использовать переборный алгоритм, вычисляющий сумму для всех возможных вариантов, поскольку написанная по такому алгоритму программа будет выполняться слишком долго.

Задание 27 традиционно считалось самым трудным заданием ЕГЭ. Задание 27 из открытого варианта можно рассматривать как некоторое обобщение используемых ранее заданий на обработку числовой последовательности, когда нужно было найти пару чисел с максимальной (минимальной) суммой, расстояние между которыми не менее чем заданное значение. В задании открытого варианта пара чисел сменилась на тройку чисел. Переборный алгоритм решения задания вполне очевиден. Эффективный алгоритм может быть построен на основе идей, используемых для эффективного алгоритма для пар чисел с применением определенной доли смекалки. Несмотря на очевидность переборного алгоритма, не все участники смогли справиться с заданием.

Анализ метапредметных результатов обучения, повлиявших на выполнение заданий КИМ ЕГЭ по информатике

При подготовке к сдаче государственной итоговой аттестации необходимо делать акцент и на метапредметные умения, влияющих на выполнение заданий КИМ ЕГЭ по информатике.

В КИМ проверяются следующие метапредметные результаты освоения основной образовательной программы:

- умение самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;

- владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания;

- готовность и способность к самостоятельной информационно-познавательной деятельности, включая умение ориентироваться в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников.

В КИМ ЕГЭ по информатике не включены задания, требующие простого воспроизведения терминов, понятий, величин, правил (такие задания слишком просты для выполнения).

При выполнении любого из заданий КИМ от экзаменуемого требуется решить тематическую задачу: либо прямо использовать известное правило, алгоритм, умение, либо выбрать из общего количества изученных понятий и алгоритмов наиболее подходящее и применить его в известной или новой ситуации.

Согласно анализу кодификатора и спецификации КИМ ЕГЭ и федеральной рабочей программы среднего общего образования «Информатика», была составлена таблица с перечнем метапредметных умений, которые могли повлиять на выполнение

задания, в том числе познавательные, коммуникативные, регулятивные (самоорганизация и самоконтроль).

№ задания в КИМ	Элементы содержания, проверяемые заданиями экзаменационной работы	Требования к результатам освоения основной образовательной программы среднего общего образования, проверяемые заданиями экзаменационной работы	Перечень метапредметных умений, которые могли повлиять на выполнение задания, в том числе познавательные, коммуникативные, регулятивные
1. Информация и кодирование			
4	Процесс передачи информации, источник и приёмник информации. Сигнал, кодирование и декодирование. Искажение информации	Интерпретировать результаты, получаемые в ходе моделирования реальных процессов	Работа с информацией. Базовые логические действия - овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-следственных связей, построение рассуждений, отнесения к известным понятиям. Базовые исследовательские действия - формулировка цели, учет изменения объекта, ситуации; проведение несложных исследований; формулировка выводов и подкрепление их доказательствами; прогнозирование возможного развития процессов, событий и т.п.
7	Форматы представления графических и звуковых объектов	Оценивать скорость передачи и обработки информации	Работа с информацией. Базовые логические действия - овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-следственных связей, построение рассуждений, отнесения к известным понятиям. Коммуникативные универсальные учебные действия - умение правильно интерпретировать условие задачи, обоснованно выбрать структуры данных и виды алгоритмов.
8	Формализация понятия алгоритма	Читать и отлаживать	Работа с информацией. Базовые логические действия -

		программы на языке программирования	овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-следственных связей, построение рассуждений, отнесения к известным понятиям.
11	Дискретное (цифровое) представление текстовой, графической, звуковой информации и видеоинформации. Единицы измерения количества информации	Оценивать объём памяти, необходимый для хранения информации	Работа с информацией. Базовые логические действия - овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-следственных связей, построение рассуждений, отнесения к известным понятиям.
2. Моделирование			
1	Описание (информационная модель) реального объекта и процесса, соответствие описания объекту и целям описания. Схемы, таблицы, графики, формулы как описания	Интерпретировать результаты, получаемые в ходе моделирования реальных процессов	Работа с информацией. Базовые логические действия - овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-следственных связей, построение рассуждений, отнесения к известным понятиям. Базовые исследовательские действия - формулировка цели, учет изменения объекта, ситуации; проведение несложных исследований; формулировка выводов и подкрепление их доказательствами; прогнозирование возможного развития процессов, событий и т.п.
13	Описание (информационная модель) реального объекта и процесса, соответствие описания объекту и целям описания. Схемы, таблицы, графики, формулы как описания	Использовать готовые модели, оценивать их соответствие реальному объекту и целям моделирования	
22	Программная и аппаратная организация компьютеров и компьютерных систем. Виды программного обеспечения	Оценивать скорость передачи и обработки информации	

3. Системы счисления			
14	Позиционные системы счисления	Строить информационные модели объектов, систем и процессов в виде алгоритмов	Работа с информацией - нахождение нужной информации, распознавание достоверной и недостоверной информации, анализ текстовой, графической, информации в соответствии с учебной задачей; самостоятельно построение схемы, таблицы для представления информации.
4. Основы логики			
2	Высказывания, логические операции, кванторы, истинность высказывания	Строить и анализировать таблицы истинности для логического высказывания	Работа с информацией. Базовые логические действия - овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-следственных связей, построение рассуждений, отнесения к известным понятиям.
15	Высказывания, логические операции, кванторы, истинность высказывания	Вычислять логическое значение сложного высказывания по известным значениям элементарных высказываний	
19	Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы)	Строить информационные модели объектов, систем и процессов в виде алгоритмов	
20	Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы)	Строить информационные модели объектов, систем и процессов в виде алгоритмов	
21	Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы)	Строить информационные модели объектов, систем и процессов в виде алгоритмов	
5. Алгоритмы и программирование			
5	Построение алгоритмов и практические вычисления	Строить информационные модели объектов, систем и процессов в виде алгоритмов	Работа с информацией. Базовые логические действия - овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-

			<p>следственных связей, построение рассуждений, отнесения к известным понятиям.</p> <p>Базовые исследовательские действия - формулировка цели, учет изменения объекта, ситуации; проведение несложных исследований; формулировка выводов и подкрепление их доказательствами; прогнозирование возможного развития процессов, событий и т.п.</p>
6	<p>Основные конструкции языка программирования.</p> <p>Система программирования</p>	<p>Читать и отлаживать программы на языке программирования</p>	<p>Работа с информацией.</p> <p>Базовые логические действия - овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-следственных связей, построение рассуждений, отнесения к известным понятиям.</p>
12	<p>Вычислимость.</p> <p>Эквивалентность алгоритмических моделей</p>	<p>Строить информационные модели объектов, систем и процессов в виде алгоритмов</p>	<p>Работа с информацией.</p> <p>Базовые исследовательские действия - формулировка цели, учет изменения объекта, ситуации; проведение несложных исследований; формулировка выводов и подкрепление их доказательствами; прогнозирование возможного развития процессов, событий и т.п.</p> <p>Коммуникативные универсальные учебные действия - умение правильно интерпретировать условие задачи, обоснованно выбрать структуры данных и виды алгоритмов.</p>
16	<p>Индуктивное определение объектов</p>	<p>Строить информационные модели объектов, систем и процессов в виде алгоритмов</p>	<p>Работа с информацией.</p> <p>Базовые исследовательские действия - формулировка цели, учет изменения объекта, ситуации; проведение несложных исследований; формулировка выводов и подкрепление их доказательствами; прогнозирование возможного развития процессов, событий и т.п.</p>
17	<p>Основные конструкции языка программирования.</p>	<p>Создавать программы на языке</p>	<p>Работа с информацией.</p> <p>Коммуникативные универсальные учебные действия - умение</p>

	Система программирования	программирования по их описанию	правильно интерпретировать условие задачи, обоснованно выбрать структуры данных и виды алгоритмов.
23	Вычислимость. Эквивалентность алгоритмических моделей	Строить информационные модели объектов, систем и процессов в виде алгоритмов	Работа с информацией. Базовые исследовательские действия - формулировка цели, учет изменения объекта, ситуации; проведение несложных исследований; формулировка выводов и подкрепление их доказательствами; прогнозирование возможного развития процессов, событий и т.п.
24	Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы)	Строить информационные модели объектов, систем и процессов в виде алгоритмов	Работа с информацией. Коммуникативные универсальные учебные действия - умение правильно интерпретировать условие задачи, обоснованно выбрать структуры данных и виды алгоритмов.
25	Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы)	Строить информационные модели объектов, систем и процессов в виде алгоритмов	Работа с информацией. Коммуникативные универсальные учебные действия - умение правильно интерпретировать условие задачи, обоснованно выбрать структуры данных и виды алгоритмов.
26	Построение алгоритмов и практические вычисления	Строить информационные модели объектов, систем и процессов в виде алгоритмов	Работа с информацией. Коммуникативные универсальные учебные действия - умение правильно интерпретировать условие задачи, обоснованно выбрать структуры данных и виды алгоритмов.
27	Построение алгоритмов и практические вычисления	Строить информационные модели объектов, систем и процессов в виде алгоритмов	Работа с информацией. Коммуникативные универсальные учебные действия - умение правильно интерпретировать условие задачи, обоснованно выбрать структуры данных и виды алгоритмов.
6. Электронные таблицы			
3	Системы управления базами данных. Организация баз данных	Создавать и использовать структуры хранения данных	Работа с информацией. Базовые логические действия - овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-следственных связей, построение

			рассуждений, отнесения к известным понятиям.
9	Умение обрабатывать числовую информацию в электронных таблицах. Обработка статистических данных	Представлять и анализировать табличную информацию в виде графиков и диаграмм	Работа с информацией. Базовые логические действия - овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-следственных связей, построение рассуждений, отнесения к известным понятиям.
18	Умение использовать электронные таблицы для обработки целочисленных данных. Использование динамических (электронных) таблиц для выполнения учебных заданий из различных предметных областей	Представлять и анализировать табличную информацию в виде графиков и диаграмм	Работа с информацией. Базовые логические действия - овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-следственных связей, построение рассуждений, отнесения к известным понятиям.
7. Информационный поиск			
10	Использование инструментов поисковых систем (формирование запросов)	Осуществлять поиск и анализ информации в реляционных базах данных	Работа с информацией. Базовые логические действия - овладения действиями сравнения, анализа, синтеза, обобщения, классификации по признакам, установление аналогий и причинно-следственных связей, построение рассуждений, отнесения к известным понятиям.

Примечание:

Работа с информацией - нахождение нужной информации, распознавание достоверной и недостоверной информации, анализ текстовой, графической, информации в соответствии с учебной задачей; самостоятельно построение схемы, таблицы для представления информации. Такая компетенция, как работа с информацией, является одной из ключевых базовых компетенций. Для решения любой задачи необходимо внимательно ознакомиться с информацией, которая дана в условии задачи, понять ее содержание (смысловое чтение), проанализировать ее. В КИМ ЕГЭ по информатике имеется ряд заданий с объемными формулировками условий, при решении которых обучающиеся, к сожалению, допускают много ошибок, что связано с их неумением работать с текстом задачи.

Регулятивные универсальные учебные действия - определение проблемы, цели; планирования своей деятельности, нахождение алгоритма решения, выдвижения

гипотезы, оформление, проверка и оценка конечного результата, корректировка, самостоятельная работа с информацией для выполнения конкретного задания, проведение анализа проделанной работы и выводы. В данном случае, это самоорганизация и самоконтроль. Настрой на успешное выполнение заданий КИМ ЕГЭ, и проверка полученных результатов гарантирует достижение хороших результатов. Основным показателем достижения результатов является освоение учащимися средств управления своей учебной деятельностью. К этому следует относиться как к обязательной части выполнения любого задания. Для эффективного самоанализа и самоконтроля в течение учебного года и в ходе подготовки к ЕГЭ необходимо знакомить учащихся с подробной инструкцией для оценивания заданий. Это помогает понять аргументированность оценки, определить пробелы в знаниях.

Необходимо отметить, что ключевым фактором выполнения заданий ЕГЭ по информатике является сформированность метапредметных навыков, относящихся прежде всего к универсальными учебными познавательными действиями и учебными регулятивными действиями:

1. базовые логические действия, таких как способность самостоятельно выбирать способ решения учебной задачи (сравнивать несколько вариантов решения, выбирать наиболее подходящий с учетом самостоятельно выделенных критериев).

2. навыки работы с информацией, такие как: выбирать, анализировать, систематизировать и интерпретировать информацию различных видов и форм представления; самостоятельно выбирать оптимальную форму представления информации и иллюстрировать решаемые задачи несложными схемами, диаграммами, иной графикой и их комбинациями; эффективно запоминать и систематизировать информацию.

3. 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

В качестве примера недостаточной сформированности метапредметных умений, навыков и способов деятельности можно представить такие задания:

Задания 12 и 18. На примере этих заданий, которые рассматривались выше, можно сделать вывод, что низкий показатель решения свидетельствует о недостаточной сформированности таких метапредметных результатов, как владение грамотной устной и письменной речью, выстраивание цепочек аналогий и взаимосвязей, создание моделей и схем, при помощи которых можно выполнять познавательные задачи.

Такие метапредметные обобщенные способы действий, как схематизация, моделирование оказались недостаточно сформированы, что наблюдается и при решении задания 26, о котором тоже говорилось выше.

Недостаточно ориентируясь в предоставляемой информации, учащиеся опирались на ожидаемые формулировки заданий, стремясь минимизировать усилия, решали задачи, отработанные на уроках. Примером может служить задание 6.

Необходимо продолжить работу по формированию умения построения логически верных рассуждений и подтверждения их истинности. Для решения задач 24–27 необходимо критическое мышление и креативный подход, который можно сформировать у выпускников только при системном подходе.

Участниками экзамена при выполнении заданий базового и повышенного уровней сложности был продемонстрирован наиболее высокий уровень форсированности следующих знаний и умений:

- 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;
- знание о позиционных системах счисления и двоичном представлении информации в памяти компьютера;
- умение осуществлять информационный поиск средствами операционной системы или текстового процессора;
- умение анализировать алгоритм логической игры;
- умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд;
- знание позиционных систем счисления;
- вычисление рекуррентных выражений;
- умение найти выигрышную стратегию игры;
- умение анализировать алгоритм, содержащий ветвление и цикл.

У участников ЕГЭ 2023 года возникли затруднения при выполнении заданий повышенного и высокого уровней сложности, контролирующими следующие знания и умения:

- 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;
- знание позиционных систем счисления;
- знание основных понятий и законов математической логики;
- умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования;
- умение использовать электронные таблицы для обработки целочисленных данных;
- умение анализировать результат исполнения алгоритма, содержащего ветвление и цикл;
- умение создавать собственные программы (10–20 строк) для обработки символьной информации, обработки целочисленной информации;
- умение обрабатывать целочисленную информацию с использованием сортировки;

- умение создавать собственные программы (20–40 строк) для анализа числовых последовательностей.

- умение создавать собственные эффективные программы и анализировать их трудоемкость.

Рекомендации для системы образования Томской области

Рекомендации по совершенствованию преподавания учебного предмета всем обучающимся

Учителям, методическим объединениям учителей:

Результаты ГИА-11 по информатике позволяют сформулировать некоторые общие рекомендации, направленные на совершенствование организации и методики преподавания предмета «Информатика»:

1) Подготовка к экзамену должна осуществляться не только в ходе массированного решения вариантов - аналогов экзаменационных работ, а, в основном, в ходе грамотно организованного учебного процесса, в результате которого у обучающихся формируются необходимые предметные и метапредметные компетенции.

Необходимо выключить в процесс обучения следующие технологии для формирования метапредметных компетенций: технологии совместного обучения; технологии исследовательской деятельности; проектной деятельности; проблемно-диалогической технологии; игровые технологии и другие.

2) Рациональным подходом к подготовке обучающихся к ЕГЭ является систематическое изучение теоретического материала по каждой теме, рассмотрение всевозможных методов решения различных типов задач, их отработка путём решения большого количества заданий. Причём подготовку к итоговой аттестации следует проводить на протяжении всего периода обучения в школе.

3) Необходимо внедрение эффективных механизмов текущего и тематического контроля на школьном уровне, что даст возможность отслеживать результаты обучающихся по наиболее важным темам курса, через различный вид диагностических и проверочных работ, и своевременно корректировать уровень усвоения изучаемого материала.

4) Необходимо заранее познакомить обучающихся с критериями оценивания работ ЕГЭ. В процессе обучения также следует оценивать диагностические работы, следуя критериям ЕГЭ.

5) В связи с проведением ЕГЭ по информатике в компьютерной форме, целесообразно на всех уровнях общего образования при изучении информатики уделять особое внимание решению задач, в том числе и по теоретической информатике, с использованием компьютерных инструментов: средств программирования, электронных таблиц, текстового процессора. Причём обучение прикладным программам рекомендуется проводить без привязки к конкретному ПО, рассматривать несколько пакетов офисных программ.

6) Важно отметить, что тенденцию по переходу на более современные языки программирования (например, Python), необходимо продолжить в новом учебном году.

7) При подготовке обучающихся к ЕГЭ следует обратить особое внимание на прочное усвоение теоретических основ информатики, с учетом тесных межпредметных связей информатики с математикой, увеличение количества практических занятий, развитие метапредметных умений обучающихся. Использовать на уроках информатики задания, для выполнения которых необходимо применять устный счет и математический аппарат, так как на результаты выполнения экзаменационной работы существенно влияет уровень общей математической подготовки выпускников. В целях повышения качества метапредметных результатов обучения рекомендуется применять

различные методы обучения: метод проблемного обучения, исследовательский метод, метод кейс-стади и т.д.

8) При изучении информатики важно обратить самое пристальное внимание на анализ и программирование алгоритма для заданного исполнителя; знание основных управляющих структур языка программирования; умение разрабатывать модель решения задачи; умение выполнять формализацию и постановку решения задачи, умение определять количество и информационный объём файлов, отобранных по некоторому правилу, а также на усвоение умения проводить обработку массива целочисленных данных с использованием средств электронной таблицы, работать в прикладных программах.

9) При изучении темы «Алгоритмы и исполнители» важно познакомить обучающихся с визуальными средами программирования, например, «Кумир» с первого года изучения информатики; рассмотреть основные алгоритмические конструкции: следование, ветвление, цикл; научить записывать алгоритм с помощью блок-схем; выделить больше времени на анализ готовых алгоритмов.

10) При изучении темы «Основы программирования» необходимо при знакомстве с языком программирования уделить особое внимание понятиям «величина» и типы величин, «команда присваивания»; рассматривать как простые, так и составные условия с использованием логических операций в полном и неполном ветвлении; изучить цикл с заданным количеством повторений и цикл с параметром; изучить арифметические операции выделения целой части и остатка от деления для целых чисел; познакомить обучающихся с алгоритмами поиска экстремальных значений (без использования встроенных функций); при решении задач уделить больше внимания формализации, построению математической модели.

11) При изучении темы «Представление и обработка информации в электронных таблицах» следует научить формулировать краткое условие задачи, выделять, что дано и что нужно найти; познакомить обучающихся с разными видами сортировки массивов данных, научить пользоваться фильтрами; научить пользоваться встроенными математическими и статистическими функциями для обработки диапазона ячеек электронной таблицы, такими как: сумма, среднее значение, минимальное и максимальное значение, количество элементов, отвечающих заданному условию; обязательно рассматривать запись условной функции и базовых логических операций.

12) Важно проводить практические работы (с использованием и без использования компьютера) по определению информационного объема информации различных типов.

13) Формировать с помощью тренингов или практикумов, самоконтроля практический опыт работы для вычисления выражений (с использованием и без использования компьютера), выполнения комплексных заданий межпредметного характера. Необходимо предлагать учащимся решать и оценивать по критериям решения практических заданий, образующих систему заданий: задания на работу в различных программных средах; задания на применение одного условия и задачи на комплексное применение знаний; задания на формализацию условий задач с одним или несколькими условиями; задания на изучение различных способов решения заданий.

14) Необходимо использовать предметную и метапредметную проектную деятельность, особенно для выработки навыков алгоритмизации и программирования.

15) Важно рационально сочетать различные приемы и методы, используемые на уроке, направленные на организацию самостоятельной деятельности каждого обучающегося; при этом непременным условием является проведение мероприятий по формированию навыков самоконтроля и самопроверки выполненных учеником заданий,

что способствует повышению качества выполняемой работы и формированию личной ответственности обучающегося за свои собственные результаты обучения.

16) Использовать в качестве методической поддержки материалы с сайта ФИПИ: документы, определяющие структуру и содержание контрольных измерительных материалов (кодификатор элементов содержания, спецификация, демонстрационные варианты контрольных измерительных материалов); открытый сегмент Федерального банка тестовых заданий.

Муниципальным органам управления образованием:

Провести анализ результатов ЕГЭ, выявить перечень образовательных организаций (ОО), продемонстрировавших наиболее высокие и низкие результаты ЕГЭ по учебному предмету.

Обеспечить обмен практиками ОО, продемонстрировавших наиболее высокие результаты ЕГЭ с ОО, продемонстрировавшими низкие результаты по учебному предмету при подготовке к ЕГЭ года с целью повышения результатов последних.

Обеспечить участие учителей в мероприятиях разного уровня, посвященных подготовке к ЕГЭ по информатике.

Спланировать на муниципальном уровне системную методическую поддержку непрерывного профессионального роста учителя (наставничество, «горизонтальная кооперация», «школа молодого учителя» и др.).

Прочие рекомендации:

В целом работу учителя по подготовке обучающихся к ЕГЭ рекомендуется строить по следующим направлениям:

1) Работа по самообразованию и методическая работа учителя.

С организационной точки зрения учителю необходимо:

- изучить нормативно-правовые документы;

- изучить пакет КИМ;

- изучить статистико-аналитический отчет и методический анализ результатов государственной итоговой аттестации. Особое внимание уделить изучению, представленному на сайте ФИПИ пакету КИМ ЕГЭ в новом формате, проанализировать предполагаемые изменения в КИМ;

- выстроить для обучающихся индивидуальные траектории подготовки, чтобы владеть полной картиной готовности к ЕГЭ каждого ученика.

С содержательной точки зрения педагогу необходимо:

- обратить особое внимание на преподавание и контроль знаний при изучении таких разделов, как «Исполнители и алгоритмы. Управление исполнителями», «Алгоритмы и элементы программирования», «Представление информации», «Кодирование информации», «Элементы комбинаторики, теории множеств и математической логики».

- подобрать средства и методы, позволяющие обеспечить дифференцированный подход обучения;

- выстроить систему контроля знаний;

- организовать обучение выпускников особенностям проведения компьютерной формы экзамена;

- усилить внимание к междисциплинарным связям, в частности, – математики и физики.

Использовать в работе с обучающимися учебное пособие по информатике и ИКТ «Информатика. Углубленный уровень». Согласно федеральному перечню учебников, допущенных к использованию при реализации госпрограмм начального, основного и

среднего общего образования (Приказ Минпросвещения России от 21.09.2022 № 858 (ред. от 21.02.2024)).

Рекомендации по организации дифференцированного обучения школьников с разными уровнями предметной подготовки

Учителям, методическим объединениям учителей:

При преподавании информатики и ИКТ необходимо организовывать дифференцированное обучение с учетом индивидуальных возможностей и способностей обучающихся. На уроках информатики и ИКТ рекомендуется организовать дифференцированное обучение через:

- 1) содержание учебного материала (разным группам ребят предлагают для усвоения разные учебные сведения);
- 2) разные виды деятельности;
- 3) выполнение работ разной степени сложности, но одного вида;
- 4) индивидуальные образовательные маршруты обучающихся.

При организации дифференцированного обучения по подготовке к сдаче ЕГЭ учителю необходимо выполнить следующие действия:

- 1) анализ результатов текущей, тематической, промежуточной, итоговой оценки с целью выявления типичных затруднений обучающихся;
- 2) изучение индивидуальных особенностей для составления индивидуального плана работы.

Обучающимся рекомендуется провести самодиагностику, порешав демонстрационный вариант КИМ, заполнить таблицу для оценки уровня подготовленности к ЕГЭ (данная таблица приведена в рекомендациях по самостоятельной подготовке на сайте ФИПИ: <https://fipi.ru/navigator-podgotovki/navigator-ege#inf>).

Данный механизм позволяет построить индивидуальный план подготовки к сдаче экзамена и может использоваться как учителем для формирования дифференциальной подготовки в урочной и внеурочной деятельности, так и для самостоятельной подготовки и отслеживания западающих тем учеником.

При повторении каждой темы рекомендуется сначала выполнять задания по линиям, не менее чем по три-четыре задания каждого типа, встречающегося в линии, затем выполнять задания группами, относящимися к данной теме. Обращать внимание на разные подходы к решению заданий (аналитический способ, использование ПО и т.д.) и на различные формулировки тематических заданий. После того как ошибки в выполнении заданий по данной теме сведены к минимуму, можно переходить к проработке следующей темы.

После завершения повторения всех тем следует решить ещё как минимум один вариант КИМ и сравнить результаты с предыдущими. Также снова следует выявить темы и линии заданий, вызвавшие затруднения, и дополнительно их проработать.

При изучении тем в условиях классно-урочной системы необходимо создавать по каждой теме системы задач и практических работ, имеющих последовательно возрастающую сложность.

Учителю при начале изучения каждой темы следует определить зону ближайшего развития каждого ученика и совместно с обучающимся определить конкретные цели изучения темы, сформулированную в виде класса задач и конкретных практических работ, которые учащийся научится решать и выполнять.

Для примера можно рассмотреть изучение темы «Электронные таблицы» (6 часов) в разделе «Информационные технологии» для 11 класса в рамках базового

изучения информатики. Согласно рабочей программы в данной теме ученики должны выполнить практические работы по анализу данных с помощью электронных таблиц.

Актуализация знаний в рамках уроков проводится, используя методический приём по выделению знакомых учащимся команд в среде электронной таблицы. Для этих целей возможно использование некоторых материалов и заданий, представленных на уроке 14 «Обработка информации в электронных таблицах» на платформе РЭШ (<https://resh.edu.ru/subject/lesson/5817/conspect/82476/>).

Данный приём нацеливает учащихся на исследовательские методы работы и вырабатывает самостоятельный поисковый стиль работы. Следует обратить внимание на переход от небольших упражнений, позволяющих вспомнить некоторые приёмы работы, изученные в рамках 7-9 классов (например, адресация, фильтр и сортировка, математические и логические функции), целью которых является именно освоение электронной таблицы как инструмента для дальнейшего использования при решении практических задач, к самим содержательным задачам.

Рассмотрение конкретных классов задач приводит к совместному с обучающимися определению конкретных целей изучения темы.

Пример задания приведен из урока «Статистическая обработка данных средствами редактора электронных таблиц» каталога цифрового образовательного контента (ФГИС «Моя школа»: Информатика 11 класс базового уровня, урок 60).

В таблице приведены некоторые статистические данные об уровне жизни Владимирской области с 2000 по 2021 года. Изучите таблицу и ответьте на вопросы.

Задание 1. Безработица

В какой год во Владимирской области был самый высокий показатель безработицы

Задание 2. Доходы

Какой средний показатель доходов во Владимирской области за 2000-2021 года.

Задание 3. Безработица 2

Сколько раз показатель безработицы поднимался выше 10%

Задание 4. Зависимости

Какие показатели имеют наибольший показатель зависимости? Это зависимость прямая или обратная?

В рамках данного примера ученики формируют цель по формированию навыка решения простых задачи анализа данных с помощью электронных таблиц. Анализируя какие подходы, использовали ученики для нахождения ответа, скорость выполнения задания, полученные результаты, преподаватель может дифференцировать учеников по уровню знаний и выстроить дальнейшую работу.

При работе с кластером подобных задач, которые выстроены в порядке последовательно возрастающей сложности, ученики изучают инструменты электронных таблиц, получают навыки по формулированию этапов работы с таблицей по заданным условиям.

Использование далее практических заданий «Оптимизация с помощью табличных процессоров» (например, практические работы к учебнику «Информатика. Углублённый уровень» для 10-11 классов К.Ю. Полякова и Е.А. Еремина, [practice10-9.doc \(live.com\)](https://practice10-9.doc.live.com)) поможет получению навыков самостоятельного формулирования алгоритма решения задач по оптимизации.

На основе предполагаемых результатов ЕГЭ по информатике обучающихся можно условно разделить на три группы: группа с низким уровнем усвоения (предполагаемые результаты экзамена – ниже минимального балла); группа со средним уровнем усвоения (предполагаемые результаты ЕГЭ – от минимального до 60 тестовых баллов); группа с высокими результатами (предполагаемые результаты от 61 до 100

тестовых баллов). Зная это, можно проводить дифференциацию при выборе информационных задач и разнообразных методов/приемов обучения.

В работе с обучающимися *с уровнем подготовки ниже среднего* необходима работа с базовыми информационными понятиями и конструкциями. В первую очередь нужно обращать внимание при подготовке на формирование умений определения возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов, обрабатывать числовую информацию в электронных таблицах, 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, знание основных понятий и методов, используемых при измерении количества информации.

Вторая многочисленная группа обучающихся *со средним уровнем подготовки* нуждается в дополнительной работе с алгоритмическим и программируемым материалом, выполнении большого количества различных заданий, предполагающих преобразование и интерпретацию информации. Приоритетной технологией здесь может стать совместное обучение – *технология сотрудничества*.

На занятиях можно применять разные варианты технологии сотрудничества. Так, например, вариант «обучение в команде»: при изучении систем счисления ученики разбиваются на группы в четыре - пять человек (обязательно разные по уровню обученности). Учитель объясняет новый материал, а затем предлагает ученикам в группах его закрепить, постараться разобраться, понять все детали. Перевод заданных чисел по цепочке: из десятичной системы счисления (СС) в двоичную СС, затем результат перевести в восьмеричную СС, далее в шестнадцатеричную СС и последнее число переводится в десятичную СС. Таким образом, каждый ученик выполняет свою часть по «вертушке» (каждое последующее задание выполняется следующим учеником, начинать может либо сильный ученик, либо слабый). При этом выполнение каждого задания объясняется каждым учеником и контролируется всей группой. После выполнения заданий всеми группами, учитель дает тест на проверку понимания нового материала.

Задания учащиеся выполняют индивидуально, вне группы. При этом учитель дифференцирует сложность заданий для сильных и слабых учеников. Оценки за выполнение индивидуальных заданий (теста) суммируются на группу и объявляется общая оценка группе. Таким образом, соревнуются не сильные со слабым, а каждый, стараясь выполнить свои задания, как бы соревнуется сам с собой, т.е. со своим ранее достигнутым результатом. И сильный, и слабый ученик могут принести группе одинаковые оценки или баллы. Это чрезвычайно эффективная работа для усвоения нового материала каждым учеником.

Таким образом, в результате совместной работы по технологии сотрудничества достигается усвоение всего материала. Основные принципы: награды всей команде, индивидуальный подход, равные возможности. Следует отметить, что при использовании технологии сотрудничества, недостаточно сформировать группы и дать задания. Суть как раз и состоит в том, чтобы учащийся захотел сам приобретать знания. А ситуация успеха группы становится фактором развития личности ученика и залогом положительного отношения к обучению, к науке, к труду, к себе. Обучение в сотрудничестве обеспечивает не только успешное освоение учебного материала всеми учениками, но и способствует их интеллектуальному развитию, активности и самостоятельности в достижении поставленной задачи.

Приоритетом в выборе методов обучения для третьей группы обучающихся *с высоким уровнем подготовки* может стать технология «перевернутого» обучения. В

процессе обучения эти школьники проявляют мотивацию к изучению информатики и, как правило, обладают достаточными знаниями для серьезной самостоятельной работы. Необходимо постоянное поддержание интереса и мотивации; развитие мышления ученика, через решение задач нестандартных и повышенной сложности, головоломок, участие в олимпиадах.

На уроке *со слабо мотивированными обучающимися* следует применять методы и приёмы работы, направленные на предупреждение неуспеваемости школьников. Для этого можно применять различные виды дифференцированной помощи: развивать устойчивый интерес к предмету; реализации различных форм и методов организации деятельности обучающихся на уроке; снижение перегрузок обучающихся, используя индивидуализацию задания; формирование умения самостоятельно работать над заданием; работа над ошибками на уроке и включение её в домашние задания; деление сложного задания на элементарные составные части.

ЕГЭ по информатике проверяется не знание синтаксиса конкретного языка программирования, а умение читать, формально исполнять, анализировать и составлять алгоритмы. Для этого школьнику необходимо обладать специфическими стилями мышления, операциональным, алгоритмическим и объектным.

Для развития данных стилей мышления можно выделить комплексы методических приемов, применение которых способствует развитию каждого из перечисленных выше стилей мышления.

Для операционального стиля мышления:

1) Трассировка – пошаговое исполнение готового алгоритма (линейного, разветвляющегося, циклического). Алгоритмы должны быть представлены в различных формах (словесное описание, блок-схема, язык программирования).

2) Построение формулировки задачи, которую решает представленный алгоритм.

3) Поиск и исправление синтаксических ошибок в алгоритме.

4) Поиск и исправление логических (семантических) ошибок алгоритма.

5) Оптимизация готового алгоритма.

Для алгоритмического стиля мышления:

1) Создание нового алгоритма, его запись, проверка и исполнение самим обучаемым или выбранным исполнителем.

2) Усвоение алгоритмов решения основных типовых задач.

3) Поиск и исправление синтаксических и семантических ошибок в алгоритме.

4) Оптимизация готового алгоритма.

Для объектного стиля мышления:

1) Выделение объектов предметной области задачи, их статических и динамических свойств, построение иерархии объектов.

2) Построение объектной модели задачи.

3) Описание событий и поведения объектов.

Применение этих методических приемов в учебном процессе и при подготовке к экзамену по информатике позволит школьникам усвоить стили мышления, необходимые для изучения программирования и успешной сдачи экзамена, а также для выполнения заданий, не связанных с программированием или заданий с нестандартной формулировкой.

Для *мотивированных обучающихся всех уровней* рекомендуется составить каталог для самостоятельной подготовки, содержащий дополнительную литературу, расширяющую материал учебников, список онлайн-курсов, углубляющих знания не только по решению той или иной задачи, но и отдельного раздела курса информатики.

Для отработки навыков решения типовых задач можно предлагать обучающимся ресурсы, содержащие тестирующие системы: открытый банк заданий ЕГЭ ФИПИ (<https://fipi.ru/ege>); сайт «Преподавание, наука и жизнь» К.Ю. Полякова (<https://kpolyakov.spb.ru/school/ege.htm>); сайт КЕГЭ (kompege.ru) (<https://kompege.ru/>); Яндекс Учебник (<https://education.yandex.ru/ege>); Школково (<https://3.shkolково.online/catalog?SubjectId=5>).

Работа по составлению каталога может вестись постепенно в рамках методического объединения образовательного учреждения или муниципалитета.

Администрациям образовательных организаций:

Обеспечить осуществление образовательных программ по информатике базового и профильного (углубленного) уровня с использованием соответствующих учебно-методических комплексов.

Оценить возможности и найти способы использования различных форм внеурочной деятельности по предмету «Информатика» для организации дополнительной подготовки к государственной итоговой аттестации групп учащихся с разным уровнем подготовки по предмету.

Как показывают результаты ЕГЭ по предмету «Информатика», наиболее успешными являются те ОО, в которых проводится специализированное профильное обучение по предмету.

Обеспечить проведение в образовательной организации независимых диагностических процедур, направленных на выявление возможных пробелов в подготовке у потенциальных участников экзамена.

При необходимости акцентировать внимание педагогов о необходимости построения индивидуальной траектории профессионального развития по средствам индивидуального образовательного маршрута с включением мероприятий формального, неформального и информального образования регионального и муниципального уровней.

Муниципальным органам управления образованием:

Создать условия для углубленного изучения информатике в общеобразовательных организациях муниципального района, в том числе с использованием механизмов сетевого взаимодействия, дистанционного обучения.

Рекомендовать руководителям общеобразовательных организаций организовать работу по подготовке учителей информатики к использованию технологий дифференцированного обучения предмету, уделить внимание овладению учителями методик преподавания информатики как в классах с углубленным изучением, так и в классах с изучением информатики на базовом уровне.

Установить взаимодействие с ведущими региональными специалистами в области методики преподавания информатики для подготовки учителей информатики, осуществляющих дифференцированное обучение предмету, и для работы с одаренными школьниками.

Прочие рекомендации:

Обобщая вышеизложенное, выделим основные направления работы, которые необходимо учитывать при подготовке к ЕГЭ по информатике:

- усилить внимание к формированию метапредметных навыков познавательной и регулятивной направленности в контексте частичного изменения и усложнения формулировок заданий (это относится и к ученикам, и к педагогам);

- включать больше практических заданий на работу с электронными таблицами и программированием, стимулируя учеников в самостоятельной работе под руководством учителя;
- не ослаблять внимания ни к одной из тем информатики;
- активно использовать потенциал информационно-образовательного пространства, расширять информационно-образовательную среду школы;
- применять при подготовке и проведении уроков современные сервисы, обеспечивающие повышение наглядности, разные форматы подачи материала, интерактивность и оперативность при проведении различных видов оценивания;
- осваивать новые педагогические технологии для обеспечения разных вариантов включения учеников в образовательный процесс;
- постоянно заниматься самообразованием и повышением квалификации на базе региональных мероприятий и с использованием потенциала интернет-ресурсов.

Рекомендации по темам для обсуждения / обмена опытом на методических объединениях учителей информатики

Специалистам муниципальных методических служб, школьных методических объединений усилить работу по методическому сопровождению профессионального развития педагогических работников.

Провести работы по обобщению и трансляции инновационного и успешного опыта применения активных методик дифференцированного обучения информатике школьников с разным уровнем предметной подготовки.

Провести работы по соотнесению поурочных планов базовых и углубленных курсов информатики с проверяемыми на ГИА умениями и компетенциями, корректировку планов на основе проведенного анализа.

Акцентировать внимание педагогов о необходимости построения индивидуальной траектории профессионального развития посредством индивидуального образовательного маршрута с включением мероприятий формального, неформального и информального образования регионального и муниципального уровней.

Для обсуждения на методических объединениях учителей предметников могут быть рекомендованы следующие темы:

-«Обоснование выбора структуры данных для решения задачи». Здесь необходимо рассмотреть задачи, решаемые как с использованием массива, так и без него. Показать эффективные решения. Сюда относятся приемы поиска второго максимума, определение массива остатков при делении, определение массива фиксированной длины для учета расстояния между парами элементов и т.д. Необходимо ознакомиться с методикой преподавания подобной тематики. Рассматриваемые приемы можно применять для решения задания 27.

-«Методика освоения учащимися формального исполнения алгоритмов». Необходимо раскрыть приемы прочтения готовых алгоритмов, научить обучающихся навыкам генерации значений для перебора решений. Такие приемы будут полезны учащимся для успешного решения заданий 6, 15, 22. Здесь же можно рассмотреть методику построения рекурсивных алгоритмов, основанных на рекуррентных соотношениях. Разработать наборы заданий в соответствии с заданиями 6, 15, 16, 22 в формате ЕГЭ.

-«Особенности программирования на языке Python». Необходимо показать преимущества решения некоторых задач на языке Python. Показать преимущества обработки строк в этом языке, сортировку массива и решение других типовых задач,

которые в дальнейшем будут опорными для продолжения обучения в вузе и решения задач в реальной разработке программных продуктов.