



Воронежский институт высоких технологий - автономная  
некоммерческой образовательной организации высшего образования  
(ВИВТ - АНОО ВО)

УТВЕРЖДАЮ

Председатель экзаменационной  
комиссии

В.В.Куралесин

25 декабря 2025 г.

## **ПРОГРАММА ВСТУПИТЕЛЬНЫХ ИСПЫТАНИЙ**

по дисциплине «Основы информационных технологий»

для поступающих по программам бакалавриата на направления подготовки  
09.03.01 Информатика и вычислительная техника, 09.03.02 Информационные  
системы и технологии

Воронеж 2025

Программа вступительных испытаний по дисциплине «Основы информационных технологий» для поступающих на базе среднего профессионального образования соответствующего профиля по программам бакалавриата на направления подготовки 09.03.01 Информатика и вычислительная техника, 09.03.02 Информационные системы и технологии,

## СОДЕРЖАНИЕ ПРОГРАММЫ

1. Информация и её кодирование.
  - 1.1. Кодирование и декодирование информации.
  - 1.2. Единицы измерения информации.
  - 1.3. Объем памяти для хранения графической и звуковой информации.
  - 1.4. Методы, используемые при измерении количества информации.
2. Системы счисления.
  - 2.1. Позиционные системы счисления.
  - 2.2. Перевод чисел из одной системы счисления в другую.
  - 2.3. Арифметические операции в различных позиционных системах счисления.
  - 2.4. Перебор слов и системы счисления.
3. Логика.
  - 3.1. Основные понятия и законы математической логики.
  - 3.2. Основные логические операции.
  - 3.3. Построение таблиц истинности логических выражений.
  - 3.4. Проверка логических закономерностей.
  - 3.5. Логические уравнения.
  - 3.6. Логические операции и операции над множествами.
4. Анализ информационных моделей.
  - 4.1. Поиск количества путей в графе.
  - 4.2. Соотнесение таблицы и графа.
  - 4.3. Оптимизация маршрута по таблице.
  - 4.4. Построение дерева игры по заданному алгоритму и поиск выигрышной стратегии.
5. Сетевые информационные технологии.
  - 5.1. IP адреса и маска подсети.
  - 5.2. Адрес документа в Интернете.
  - 5.3. Работа с поисковыми запросами.
6. Информационные технологии.
  - 6.1. Файловая система. Поиск в файловой системе.
  - 6.2. Электронные таблицы. Вычисление суммы, среднего арифметического, наибольшего и наименьшего значений диапазона.
  - 6.3. Электронные таблицы. Работа с относительными и абсолютными ссылками. Работа с диаграммами.
  - 6.4. Табличные (реляционные) базы данных. Запросы на выборку данных. Запросы с параметрами.
7. Основы алгоритмизации.

- 7.1. Понятие алгоритма и его свойства.
- 7.2. Способы записи (описания) алгоритма: словесно-формульный, блок-схема алгоритма, псевдокод, алгоритмический язык.
- 7.3. Типовые структуры алгоритмов: алгоритмы линейной, разветвляющейся и циклической структуры.
- 7.4. Рекурсивные алгоритмы.
8. Программирование.
  - 8.1. Анализ программы с циклами и условными операторами.
  - 8.2. Обработка конечной числовой последовательности (вычисление сумм, произведений, количества элементов с заданными свойствами).
  - 8.3. Решение задач методом перебора (поиск наибольшего общего делителя двух натуральных чисел, проверка числа на простоту).
  - 8.4. Обработка символьных данных.
  - 8.5. Работа с элементами массива с однократным просмотром массива.
  - 8.6. Сортировка одномерного массива.

## **ОСНОВНЫЕ УМЕНИЯ И НАВЫКИ**

В ходе вступительного испытания абитуриент должен продемонстрировать:

### **Знания:**

- Равномерные и неравномерные коды. Декодирование сообщений, записанных с помощью неравномерных кодов. Условие Фано.
- Теоретические подходы к оценке количества информации. Закон аддитивности информации. Формула Хартли. Информация и вероятность. Формула Шеннона.
- Непрерывные и дискретные величины и сигналы. Необходимость дискретизации информации, предназначенной для хранения, передачи и обработки в цифровых системах.
- Алгоритм перевода целого числа из  $P$ -ичной системы счисления в десятичную и обратно. Арифметические операции в позиционных системах счисления.
- Кодирование текстов. Кодировка ASCII. Стандарт UNICODE.
- Логические операции. Законы алгебры логики.
- Графы. Основные понятия. Виды графов. Деревья.
- Адресация в сети Интернет. Протоколы стека TCP/IP. Разделение IP-сети на подсети с помощью масок подсетей.
- Файловые системы. Принципы размещения и именования файлов в долговременной памяти. Шаблоны для описания групп файлов.
- Анализ данных с помощью электронных таблиц.
- Табличные (реляционные) базы данных. Запросы к базе данных.
- Язык программирования (Python/C++/Pascal). Ветвления. Сложные условия. Циклы с условием. Циклы по переменной.
- Рекурсия. Рекурсивные процедуры и функции.
- Обработка символьных данных.

— Одномерные и двумерные массивы. Работа с элементами массива.

**Умения и навыки:**

- Уметь строить неравномерные коды, допускающие однозначное декодирование сообщений (префиксные коды).
- Определять объем памяти, необходимый для хранения информации.
- Подсчитывать информационный объем сообщения.
- Оценивать скорость передачи и обработки информации.
- Строить модели объектов, систем и процессов в виде таблицы истинности для логического высказывания.
- Вычислять логическое значение сложного высказывания по известным значениям элементарных высказываний.
- Владеть методами поиска информации в сети Интернет.
- Использовать маску подсети.
- Владеть навыками работы с операционными системами.
- Представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы).
- Производить вычисления в электронных таблицах, анализировать информацию, представленную в табличном виде, в виде графиков и диаграмм.
- Уметь использовать табличные (реляционные) базы данных, в частности, составлять запросы к базам данных.
- Читать и отлаживать программы на языке программирования.
- Уметь реализовывать на выбранном языке программирования высокого уровня (Pascal, Python, C++) типовые алгоритмы обработки чисел, числовых последовательностей и массивов.

**ФОРМА ПРОВЕДЕНИЯ ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ**

Вступительное испытание для абитуриентов, поступающих на все направления подготовки и специальности, проводится в форме компьютерного тестирования. Время проведения вступительного испытания 3 часа (180 минут).

## ПРИМЕРЫ ЗАДАНИЙ ПО ТЕМАМ И УРОВНЮ СЛОЖНОСТИ

Задания на три балла имеют 4 варианта ответа, один из которых правильный

1.

Задание на 3 балла. Перевод целых чисел из 10-чной системы счисления в 2-чную, 8-чную, 16-чную и обратно.

Произведите вычисления в десятичной системе счисления:

$$(5 * 2) + (4 : 2) =$$

Ответ запишите в двоичной системе счисления.

**1100**

0011

11

00

### Решение

1) Произвести вычисление в десятичной системе  $(5 * 2) + (4 : 2) = 10 + 2 = 12_{10}$

2) Произвести перевод  $12_{10} = 1100_2$

$$\begin{array}{r|l} 12 & 2 \\ \hline -12 & 6 \quad 2 \\ \hline 0 & -6 \quad 3 \quad 2 \\ & 0 \quad -2 \quad 1 \\ & & 1 \end{array}$$

2.

Задание на 3 балла. Арифметические операции в различных позиционных системах счисления.

Решите уравнение в восьмеричной системе счисления. Значение  $x$  должно быть записано в восьмеричной системе счисления.

$$x - 413,5 = 431,1$$

**1044,6**

844,6

144,6

A44,6

### Решение

1) Выразить из уравнения  $x = 431,1 + 413,5$

2) Произвести сложение  $x = 1044,6$  (в восьмеричной системе счисления)

3.

Задание на 3 балла. Перебор слов и системы счисления.

Все 5-<sup>ю</sup> буквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке. Вот начало списка:

1. ААААА
2. ААААО
3. ААААУ
4. АААОА

.....

Запишите слово, которое стоит на 210-<sup>м</sup> месте от начала списка.

УУУАО

**УОУАУ**

ААООУ

УУААА

### Решение

Заменим буквы А, О, У на 0, 1, 2.

Выпишем начало списка, заменив буквы на цифры.

1. 00000
2. 00001
3. 00002
4. 00010

...

Полученная запись есть числа, записанные в троичной системе счисления в порядке возрастания. Тогда на 210 месте будет стоять число 209 (так как первое число — 0). Переведём число 209 в троичную систему (деля и снося остаток справа налево):

$$209 / 3 = 69 (2)$$

$$69 / 3 = 23 (0)$$

$$23 / 3 = 7 (2)$$

$$7 / 3 = 2 (1)$$

$$2 / 3 = 0(2)$$

В троичной системе 209 запишется как 21202. Произведём обратную замену и получим УОУАУ.

4.

Задание на 3 балла. Определение количества информации.

В коробке лежат 8 черных и 24 красных кубика. Сколько информации несет в себе сообщение о том, что достали кубик черного цвета?

4

**2**

8

16

**Решение**

Формула Шеннона:  $H = \log_2 \left( \frac{1}{p} \right)$ , где  $H$  – количество информации в сообщении о событии  $p$ ,  $p$  — вероятность события.

1) Подсчитаем общее количество кубиков:  $K = K_{(\text{черных})} + K_{(\text{красных})}$ .

$$K = 8 + 24 = 32.$$

2) Найдем вероятность того, что достали черный кубик:  $p = 8/32 = 1/4$

3) Произведем расчет  $H = \log_2 \frac{1}{1/4} = \log_2 4 = 2$  бита

5.

Задание на 3 балла. Кодирование и декодирование информации.

По каналу связи передаются сообщения, содержащие только восемь букв: А, Б, Г, Е, И, М, Р, Т. Для передачи используется двоичный код, удовлетворяющий условию Фано. Кодовые слова для некоторых букв известны:

Буква	Код
А	11
Б	0010
Г	1011
Е	0011
И	?
М	01
Р	000
Т	1010

Укажите кратчайшее кодовое слово для буквы И. Если таких кодов несколько, укажите код с наименьшим числовым значением.

100

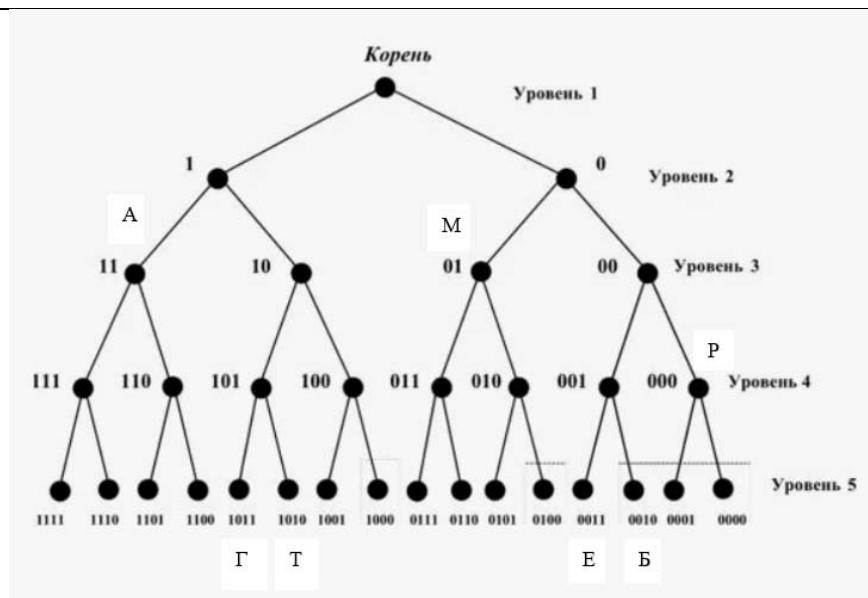
0

010

111

**Решение**

Для выполнения задания можно изобразить дерево.



Проверим каждый из 4-х вариантов ответа:

0 – нельзя, Б, Е, М и Р начинаются с 0.

010 – нельзя из-за М.

111 – нельзя из-за А.

100 – можно использовать.

6.

Задание на 3 балла. Определение информационного объема сообщения.

В одной из кодировок Unicode каждый символ кодируется 16 битами.

Определите размер следующего предложения:

«Делу время – потехе час»

Ответ представить в байтах.

46

**50**

70

43

**Решение**

1) Количество символов в предложении 25, то есть  $K = 25$ .

2)  $I = K * i = 25 * 16 = 400$  бит = 50 байт.

7.

Задание на 3 балла. Дискретное представление различных видов информации.

Кирилл делает фотографии цифровой камерой своего смартфона. Каждая фотография представляет собой растровое изображение размером  $1024 \times 768$  пикселей, при этом используется палитра из  $2^{30}$  цветов. В конце дня он отправляет снимки друзьям с помощью приложения-мессенджера. Для



экономии трафика приложение оцифровывает снимки повторно, используя размер  $800 \times 600$  пикселей и глубину цвета 28 бит. Сколько Мбайт трафика экономится при передаче 10 фотографий?

В ответе укажите целую часть до запятой.

**12**

12393

13691

13

**Решение**

1) Определим размер 10 фото исходного качества:  $V_{исх} = I * X * Y * K$ , где  $I$  – глубина цвета,  $X \times Y$  – размеры изображения,  $K$  – количество фото.

Найдем  $I$  из формулы:  $N = 2^I, 2^{30} = 2^I, I = 30$

$V = 30 * 1024 * 768 * 10 = 235\,929\,600$  бит

2) Определим размер 10 фото после сжатия:  $V_{сж} = 28 * 800 * 600 * 10 = 134\,400\,000$  бит

3) Найдем разницу:  $V = V_{исх} - V_{сж}$

$V = 235\,929\,600 - 134\,400\,000 = 101\,529\,600$  бит =  $12\,691\,200$  байт =  $12\,393,75$  Кб =  $12,10$  Мб

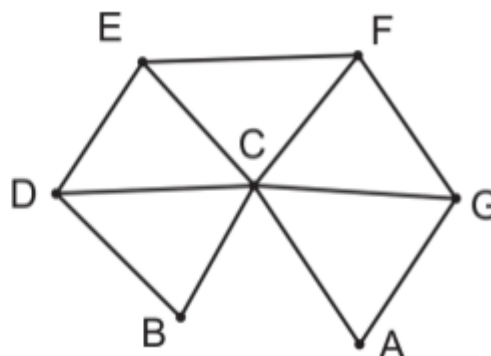
8.

Задание на 3 балла. Соотнесение таблицы и графа.

На рисунке изображена схема дорог N-ского района. В таблице звёздочкой обозначено наличие дороги из одного населённого пункта в другой.

Отсутствие звёздочки означает, что такой дороги нет.

		Номер пункта						
		1	2	3	4	5	6	7
Номер пункта	1		*	*	*	*	*	*
	2	*						*
	3	*				*		*
	4	*				*	*	
	5	*		*	*			
	6	*			*			
	7	*	*	*				



Каждому населённому пункту на схеме соответствует номер в таблице, но неизвестно, какой именно номер. Определите, какие номера в таблице могут соответствовать населённым пунктам E и F на схеме. В ответе запишите эти два номера через запятую.

**3,5**

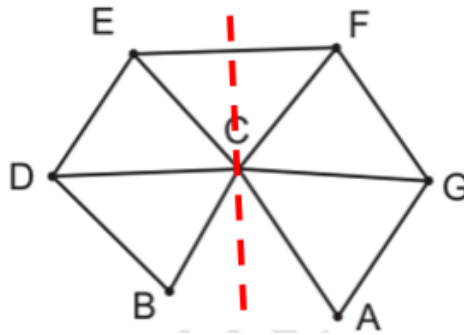
4,5

2,3

6,7

**Решение**

Заметим, что граф симметричный. Ось симметрии можно провести следующим образом:



Населенные пункты E и F расположены относительно симметрично относительно проведенной оси. Начнем восстанавливать вершины парами. Вершины A и B имеют степень 2 (пункты 2 и 6), с ними связаны C со степенью 6 (пункт 1) и D/G со степенью 3 (пункты 4 и 7). Оставшаяся пара вершин соответствует пунктам 3 и 5.

9.

Задание на 3 балла. Адресация в электронных таблицах.

Дан фрагмент электронной таблицы:

	A	B	C
1	4	2	
2	=2*(A1-C1)	=(2*B1+A1)/4	=C1-1

Какое целое число должно быть записано в ячейке C1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:C2 соответствовала рисунку?



3

4

1

5

**Решение**

По рисунку видно, что значения всех трёх ячеек диапазона A2:C2 равны. По данным таблицы найдём значение B2:  $B2 = (2 * 2 + 4) / 4 = 2$ .

Выразим  $C_1$  из значения  $C_2$ :  $C_1 = C_2 + 1 = B_2 + 1 = 2 + 1 = 3$ .

10.

Задание на 3 балла. Анализ информационных моделей.

Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.

	A	B	C	D	E	F
A		2	4	8		16
B	2			3		
C	4			3		
D	8	3	3		5	3
E				5		5
F	16			3	5	

Определите длину **КРАТЧАЙШЕГО** пути между пунктами A и F, проходящего через пункт E и не проходящего через пункт B. Передвигаться можно только по указанным дорогам.

18

17

15

16

**Решение**

Найдём все варианты маршрутов, проходящих через пункт E и не проходящих через пункт B, и выберем самый короткий.

A–D–E–F. Длина маршрута  $8 + 5 + 5 = 18$ .

A–C–D–E–F. Длина маршрута  $4 + 3 + 5 + 5 = 17$ .

Кратчайший путь равен 17.

11.

Задание на 3 балла. Обработка массивов и матриц.

В программе используется фрагмент одномерного целочисленного массива A с индексами от 1 до 10. Значения элементов равны 7; 3; 4; 8; 6; 9; 5; 2; 0; 1 соответственно, т. е.  $A[1] = 7$ ;  $A[2] = 3$  и т. д.

Определите значение переменной j после выполнения следующего фрагмента программы, записанного ниже на разных языках программирования.

Pascal	Python
<pre> j := 0; for k := 2 to 10 do begin if A[k] &lt;= A[1] then begin A[1] := A[k]; j := j + k end end; </pre>	<pre> j = 0 for k in range(2, 11): if A[k] &lt;= A[1]: A[1] = A[k] j = j + k </pre>
C++	Алгоритмический
<pre> j = 0; for (k = 2; k &lt;= 10; k++) { if (A[k] &lt;= A[1]) { A[1] = A[k]; j = j + k; } } </pre>	<pre> j := 0 НЦ для k от 2 до 10 если A[k] &lt;= A[1] то A[1] := A[k] j := j+k все КЦ </pre>

19

17

20

15

### Решение

Команда « $A[1] := A[k]$ » выполняется в том случае, если элемент  $k$  не больше первого элемента массива. Таким образом, в переменной  $j$  накапливается сумма номеров элементов, которые не больше остальных элементов, стоящих левее. Такими элементами являются:  $A[2]$ ,  $A[8]$ ,  $A[9]$ .  
Сумма их номеров: 19.

12.

Задание на 3 балла. Работа с простыми алгоритмами. Оператор присваивания, проверка на делимость.

Переменные  $x$  и  $y$  описаны в программе как целочисленные. Определите значение переменной  $x$  после выполнения следующего фрагмента программы:

<b>Pascal</b>	<b>Python</b>
<pre>x := 432; y := x div 100; x := (x mod 100) * 10; x := x + y;</pre>	<pre>x = 432 y = x // 100 x = (x % 100) * 10 x = x + y</pre>
<b>C++</b>	<b>Алгоритмический язык</b>
<pre>x = 432; y = x / 100; x = (x % 100) * 10; x = x + y;</pre>	<pre>x := 432; y := div(x,100); x := mod(x, 100) * 10 x := x + y</pre>

**324**

432

320

100

### **Решение**

Операция div делит число с отбрасыванием остатка.

Операция mod оставляет только остаток от деления числа.

$x := 432$

$y := x \text{ div } 100 = 4$

$x := (x \text{ mod } 100) * 10 = (32) * 10 = 320$

$x := x + y = 320 + 4 = 324$

13.

Задание на 3 балла. Файловая система.

Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «\*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе «\*» может задавать и пустую последовательность.

Определите, какое из указанных имён файлов удовлетворяет маске: ??pri\*.\*

**napri.q**

pripri.txt

privet.doc

3priveta.c

**Решение**

Зpriveta.c – не подходит, потому что до ргі стоит один символ.  
 privet.doc – не подходит, потому что до ргі нет ни одного символа.  
 ргірi.txt– не подходит, потому что до ргі стоят три символа.  
 Следовательно ответ: паргі.q

14.

Задание на 3 балла. Анализ и построение алгоритмов. Две линейные функции.

Запишите число, которое будет напечатано в результате выполнения программы. Для Вашего удобства программа представлена на четырех языках программирования.

Pascal	Python
<pre>var s, n: integer; begin   s := 42;   n := 1;   while s &gt; 0 do   begin     s := s - 5;     n := n + 3;   end;   writeln(n); end.</pre>	<pre>s = 42 n = 1 while s &gt; 0:     s = s - 5     n = n + 3 print(n)</pre>
C++	Алгоритмический язык
<pre>#include &lt;iostream&gt; using namespace std; int main() {   int s, n;   s = 42;   n = 1;   while (s &gt; 0) {     s = s - 5;     n = n + 3;   }   cout &lt;&lt; n &lt;&lt; endl; }</pre>	<pre>алг нач цел s, n s := 42 n := 1 нц пока s &gt; 0   s := s - 5   n := n + 3 кц вывод n кон</pre>

19

15

27

28

Решение

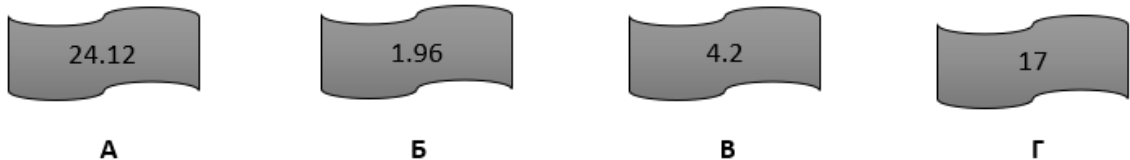
Цикл while выполняется до тех пор, пока истинно условие  $s > 0$ , т. е. переменная s определяет, сколько раз выполнится цикл. Поскольку

изначально  $s = 42$ , цикл выполнится 9 раз, следовательно,  $n = 9 \cdot 3 + 1 = 28$ .

15.

Задание на 3 балла. Восстановление IP адресов.

Даны фрагменты одного IP-адреса. Для удобства эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу. Если будет несколько вариантов решения, запишите их все через запятую.



**ГВАБ**

**ВБАГ**

**АБВГ**

**ГАБВ**

**Решение**

Для выполнения задания необходимо вспомнить основную информацию об IP-адресах.

IP-адрес в десятичном формате	192.168.31.76			
IP-адрес в десятичном формате, разбитый на октеты	192	168	31	76
IP-адрес в двоичном формате, разбитый на октеты	11000000	10101000	00011111	01001100
IP-адрес в двоичном формате	11000000	10101000	00011111	00011111

Величина октета лежит в диапазоне от 0 до 255 ( $2^i = N \Rightarrow 2^8 = N \Rightarrow N = 256$ ). Таким образом максимальным возможным числом в любом октете будет 255 (так как в двоичной системе это 8 единиц), а минимальным – 0.

**Задания на 4 балла предполагают запись ответа в поле ввода**

16.

Задание на 4 балла. Работа с IP-адресом и маской подсети.

В терминологии сетей TCP/IP IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 224.24.254.134

Маска: 255.255.224.0

В ответ запишите адрес сети.

**224.24.224.0**

**Решение**

1) Запишем маску подсети в двоичной системе счисления: 11111111  
11111111 11100000 00000000

2) Запишем IP-адрес в двоичной системе счисления: 11100000 00011000  
11111110 10000110

3) Адрес сети получается в результате поразрядной конъюнкции чисел маски и чисел адреса узла (в двоичном коде). Поскольку конъюнкция 0 с чем-либо всегда равна 0, то на тех местах, где числа маски равны 0, в адресе узла стоит 0. Аналогично там, где числа маски равны 255, стоит само число, так как конъюнкция 1 с любым числом всегда равна этому числу.

Поэтому рассмотрим только конъюнкцию числа 224 с числом 254: результат 224.

Поэтому адрес сети получим: 224.24.224.0

17.

Задание на 5 баллов. Построение таблиц истинности.

Студент заполнял таблицу истинности логической функции F

$$(x \wedge \neg y) \vee (y \equiv z) \vee \neg w,$$

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		0	0
1	0	1		0

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

В ответе запишите 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.

**wzyx**

**Решение**

Чтобы логическое сложение в результате было равно 0, нужно чтобы каждый дизъюнкт был равен 0. Или следующие три выражения должны быть справедливы  $(x \wedge \neg y) = 0$ ,  $(y \equiv z) = 0$ ,  $\neg w = 0$



x	y	$x \wedge \neg y$
0	0	0
0	1	0
1	1	0

y	z	$y \equiv z$
0	1	0
1	0	0

w	$\neg w$
1	0

Объединяем полученные наборы

x	y	z	w	F
0	0	1	1	0
0	1	0	1	0
1	1	0	1	0

Все единицы могут быть только в столбец №1. Следовательно, w – первый столбец. Заметим, что столбцы y и z являются инверсией друг друга. Таких столбца во фрагменте 2 – 2 и 3. По количеству нулей определим каждый из них. Второй столбец с двумя нулями z, третий –y. Методом исключения определим, что 4 столбец – x.

Таким образом получим ответ: wzyx

18.

Задание на 4 балла. Подсчет количества информации.

При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы из 12-символьного набора: A, B, X, D, Z, F, G, O, K, S, R, V. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей. Для хранения сведений о 20 пользователях потребовалось 400 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число — количество байт.

**12 байт**

**Решение**

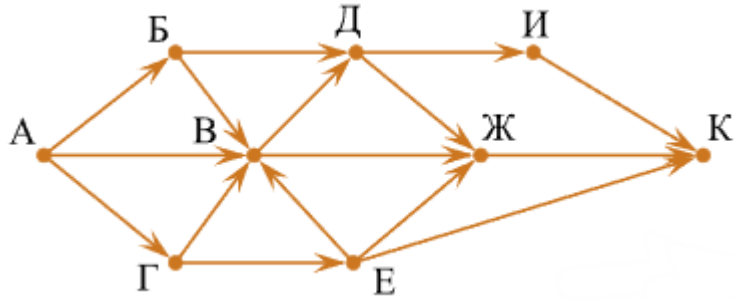
- 1) Известно, что в номере могут быть использованы 12 букв из представленного символьного набора. С помощью 1 бит можно закодировать  $2^1$  различных вариантов. Поскольку  $2^3 < 12 < 2^4$ , то для записи каждого из 12 символов необходимо 4 бита.
- 2) Для хранения всех 15 символов пароля нужно  $4 * 15 = 60$  бит, а так как для записи используется целое число байт, то берём ближайшее не меньшее значение, кратное восьми, это число  $64 = 8 * 8$  бит (8 байт).
- 3) Пусть количество памяти, отведенное под дополнительные сведения, равно x, тогда:  $20 * (8 + x) = 400$ ,  $8 + x = 400/20$ ,  $x = 20 - 8$ ,  $x = 12$ .

**Задания на 5 баллов предполагают запись ответа в поле ввода**

19.

Задание на 5 баллов. Поиск путей в графе.

На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



16

**Решение**

Один из вариантов решения – это заполнение таблицы.

Вершина	Предшествующие вершины	Количество путей
Б	А	1
В	Б+А+Г+Е	1+1+1+1=4
Г	А	1
Д	Б+В	1+4=5
Е	Г	1
И	Д	5
Ж	Д+В+Е	5+4+1=10
К	И+Ж+Е	5+10+1=16

20.

Задание на 5 баллов. Анализ алгоритмов.

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

1. Строится двоичная запись числа  $N$ .
2. К этой записи дописываются справа ещё два разряда по следующему правилу:
  - а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;
  - б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи

исходного числа  $N$ ) является двоичной записью искомого числа  $R$ .  
 Укажите минимальное число  $R$ , которое превышает 43 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

**46**

**Решение**

Если в числе было нечётное количество единиц, то в конец допишется 10.  
 Если количество единиц чётное, то допишется 00. Рассмотрим числа, большие 43. Имеем:

$44_{10} = 101100_2$  — не может являться результатом работы алгоритма;

$45_{10} = 101101_2$  — не может являться результатом работы алгоритма;

$46_{10} = 101110_2$  — может являться результатом работы алгоритма, количество единиц (кроме последних двух разрядов) нечетное, и в последних двух разрядах 10.

21.

Задание на 5 баллов. Рекурсивные алгоритмы.

Дан код на различных языках программирования, где записаны рекурсивные функции  $F$  и  $G$ . Чему будет равно значение, вычисленное при выполнении вызова  $G(5)$ ?

Алгоритмический язык	Pascal	Python
<pre> алг цел F(цел n) нач если n ≥ 2 то     знач := F(n-1)+G(n-1)+F(n-2) иначе     знач := n все кон алг цел G(цел n) нач если n ≥ 2 то     знач := G(n-1)+F(n-1)+G(n-2) иначе     знач := 3-n все кон                     </pre>	<pre> function F(n: integer): integer; begin if n &gt; 2 then     F := F(n-1)+G(n-1)+F(n-2) else     F := n; end; function G(n: integer): integer; begin if n &gt; 2 then     G := G(n-1)+F(n-1)+G(n-2) else     G := 3-n; end;                     </pre>	<pre> def F(n): if n &gt; 2: return F(n-1)+G(n-1)+F(n-2) else: return n def G(n): if n &gt; 2: return G(n-1)+F(n-1)+G(n-2) else: return 3-n                     </pre>

**26**

**Решение**

Промоделируем работу программы:  
 $G(5) = G(4) + F(4) + G(3)$ .

$$G(4) = G(3) + F(3) + G(2)$$

$$G(3) = G(2) + F(2) + G(1)$$

$$G(2) = 1$$

$$G(1) = 2$$

$$F(4) = F(3) + G(3) + F(2)$$

$$F(3) = F(2) + G(2) + F(1)$$

$$F(2) = 2$$

$$F(1) = 1$$

Теперь можно подсчитать G(3) и F(3):  $G(3) = 1 + 2 + 2 = 5$ ;  $F(3) = 2 + 1 + 1 = 4$ .

Найдём значение G(4) и F(4):  $G(4) = 5 + 4 + 1 = 10$ ;  $F(4) = 4 + 5 + 2 = 11$ .

Таким образом,  $G(5) = 10 + 11 + 5 = 26$ .

### Задания на 6 баллов предполагают запись ответа в поле ввода

22.

Задание на 6 баллов. Работа в электронных таблицах.

На рисунке приведен фрагмент таблицы с данными о чрезвычайных ситуациях. Написать следующие формулы для вычисления:

- 1) Коэффициента тяжести. Коэффициент тяжести (Низкий менее 3 человек, Средний от 3 до 5 человек, Высокий более 5 человек) зависит от количества пострадавших.
- 2) Класса аварии. Класс аварии (Локальная менее 500 тыс. руб., Серьезная от 500 тыс. до 2 млн руб., Чрезвычайная более 2 млн руб.) зависит от ущерба.
- 3) Количества аварий класса «Локальная».
- 4) Количества аварий класса «Серьезная».
- 5) Общего количества пострадавших.
- 6) Суммарного ущерба.

В ответе указывать формулы под тем номером, которому они соответствуют в задании.

	A	B	C	D	E	F	G	H	I	J	K
	№ п/п	Дата аварии	Дата ликвидации аварии	Время устарения (часы)	Сезонный период	Объект	Причина	Количество пострадавших	Коэффициент тяжести	Ущерб	Класс аварии
1											
2	8	18.01.2025	22.04.2025	2256	Зима	Цех Б	Нарушение ТБ	34		56 000 000,00 Р	
3	1	11.04.2025	15.04.2025	96	Весна	Цех А	Нарушение ТБ	7		146 000,00 Р	
4	3	13.04.2025	17.04.2025	96	Весна	Цех А	Нарушение ТБ	67		678 900,00 Р	
5	4	14.04.2025	18.04.2025	96	Весна	Цех Б	Нарушение ТБ	1		1 234 000,00 Р	
6	5	15.04.2025	19.04.2025	96	Весна	Цех А	Нарушение ТБ	90		566 700,00 Р	
7	7	17.04.2025	21.04.2025	96	Весна	Цех А	Нарушение ТБ	19		456 123,00 Р	
8	9	19.04.2025	23.04.2025	96	Весна	Цех А	Нарушение ТБ	67		123 456,00 Р	
9	10	20.04.2025	24.04.2025	96	Весна	Цех Б	Нарушение ТБ	2		789 000 000,00 Р	
10	2	12.06.2025	16.06.2025	96	Лето	Цех Б	Нарушение ТБ	12		12 000 000,00 Р	
11	6	16.12.2025	17.12.2025	24	Зима	Цех Б	Нарушение ТБ	5		12 000,00 Р	
12											
13											
14											
15											
16											
17											

1) =ЕСЛИ(Н2<3;"Низкий";ЕСЛИ(И(Н2>=3; Н2<=5);"Средний";

"Высокий"))

2) =ЕСЛИ(J2<500000;"Локальная";ЕСЛИ(И(J2>=500000; J2<=2000000);"Серьезная"; "Чрезвычайная"))

3) =СЧЁТЕСЛИ(K2:K11;"Локальная")

4) =СЧЁТЕСЛИ(K2:K11;"Серьезная")

5) =СУММ(H2:H11)

6) =СУММ(J2:J11)

### Решение

Функция ЕСЛИ в электронных таблицах позволяет выполнять логические сравнения значений и ожидаемых результатов. Пользователь задаёт условие, а программа сравнивает данные. Если условие выполняется, возвращается один результат, если нет – другой. Например, функция =ЕСЛИ(C2="Да";1;2) означает следующее: ЕСЛИ(C2="Да", то вернуть 1, в противном случае вернуть 2).

Функция И одна из логических, с помощью которой можно определить, все ли проверяемые условия принимают значение ИСТИНА. Например, =ЕСЛИ(И(A2<A3;A2<100);A2;"Значение вне допустимого диапазона.") в данном случае отображает значение в ячейке A2, если оно меньше A3 и меньше 100, в противном случае отображается сообщение "Значение выходит за пределы диапазона".

Таким образом для определения коэффициента тяжести воспользуемся двумя функциями ЕСЛИ и И. Получим формулу

=ЕСЛИ(H2<3;"Низкий";ЕСЛИ(И(H2>=3; H2<=5);"Средний"; "Высокий")).

Для определения класса аварии воспользуемся формулой

=ЕСЛИ(J2<500000;"Локальная";ЕСЛИ(И(J2>=500000; J2<=2000000);"Серьезная"; "Чрезвычайная")).

С помощью статистической функции СЧЁТЕСЛИ можно подсчитать количество ячеек, отвечающих определенному условию (например, число клиентов в списке из определенного города). Для подсчета количества аварий класса Локальная необходимо использовать формулу

=СЧЁТЕСЛИ(K2:K11;"Локальная").

Для подсчета количества аварий класса Серьезная необходимо использовать формулу

=СЧЁТЕСЛИ(K2:K11;"Серьезная").

Функция СУММ добавляет значения. Вы можете складывать отдельные значения, диапазоны ячеек, ссылки на ячейки или данные всех этих трех видов. Например, =СУММ(A2:A10) добавляет значения в ячейки A2:10.

Для определения суммарного количества пострадавших необходимо просуммировать все значения в диапазоне ячеек H2:H11. Получим формулу =СУММ(H2:H11).

Для определения суммарного ущерба необходимо просуммировать все значения в диапазоне ячеек J2:J11. Получим формулу =СУММ(J2:J11).

Задание на 6 баллов. Упрощение логических выражений.

На числовой прямой даны два отрезка:  $p = [37; 60]$  и  $q = [40; 77]$  и дано выражение:

$$(x \in p) \rightarrow (((x \in q) \cdot (x \notin a)) \rightarrow (x \notin p))$$

Укажите наименьшую возможную длину такого отрезка  $A$ , что выражение истинно при любом значении переменной  $x$ .

20

**Решение**

Обозначим логические выражения, показывающие принадлежность числа  $x$  к отрезкам:

$$P = (x \in p), \quad Q = (x \in q), \quad A = (x \in a).$$

Тогда логическое выражение, соответствующее условию задачи, может быть записано так:

$$P \rightarrow (Q \cdot \bar{A} \rightarrow \bar{P}).$$

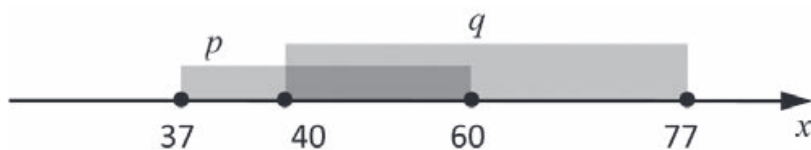
Используя свойство импликации  $A \rightarrow B = \bar{A} + B$  и закон де Моргана  $\overline{A \cdot B} = \bar{A} + \bar{B}$ , получаем:

$$P \rightarrow (Q \cdot \bar{A} \rightarrow \bar{P}) = \bar{P} + (Q \cdot \bar{A} \rightarrow \bar{P}) = \bar{P} + \bar{Q} + A + \bar{P} = A + \bar{P} + \bar{Q}.$$

В результате мы свели задачу к задаче 1, где  $B = \bar{P} + \bar{Q}$ . Её решение — условие, которое определяет минимальное множество  $A$  — получаем с помощью закона де Моргана:

$$A_{\min} = \overline{\bar{P} + \bar{Q}} = P \cdot Q.$$

Результат — это пересечение множеств  $P$  и  $Q$ , т. е. общая часть двух отрезков  $p$  и  $q$ . В нашей задаче это отрезок  $[40; 60]$ , его длина — 20



**Задания на 8 баллов предполагают запись ответа в поле ввода**

24.

Задание на 8 баллов. Исправление ошибок в программе.

Перед программистом была поставлена задача написать программу, при выполнении которой с клавиатуры считывается положительное целое число, не превосходящее  $10^9$  и которая выводит на экран максимальную цифру этого числа, кратную 3. Если в числе нет таких цифр, требуется на экран

вывести сообщение «Нет».

Программист написал программу неправильно, допустив три ошибки. Ниже приведен неверный код на трех языках программирования.

Напоминание: 0 делится на любое натуральное число.

Pascal	Python	C++
<pre>var <u>N</u>, digit, maxDigit: longint; begin   readln(A);   <u>maxDigit</u> := N mod 10;   while N &gt; 0 do   begin     <u>digit</u> := N mod 10;     if digit mod 3 = 0 then       if digit &gt; maxDigit then         <u>maxDigit</u> := digit;     <u>N</u> := N div 10;   end;   if maxDigit = 0 then     writeln('Her')   else     writeln(maxDigit) end.</pre>	<pre>A = int(input()) maxDigit = N % 10 while N &gt; 0:   digit = N % 10   if digit % 3 == 0:     if digit &gt; maxDigit:       maxDigit = digit   N = N // 10 if maxDigit == 0:   print("Her") else:   print(maxDigit)</pre>	<pre>#include &lt;iostream&gt; using namespace std; int main() {   int N, digit, <u>maxDigit</u>;   cin &gt;&gt; A;   maxDigit = N % <u>10</u>;   while (N &gt; 0)   {     digit = N % <u>10</u>;     if (digit % 3 == 0)       if (digit &gt; maxDigit)         maxDigit = <u>digit</u>;     N = N / <u>10</u>;   }   if (maxDigit == 0)     cout &lt;&lt; "Her";   else     cout &lt;&lt; maxDigit &lt;&lt; endl;   return 0; }</pre>

Требуется последовательно расписать каждое из заданий.

1) Напишите, что выведет эта программа при вводе числа 138 **ДО** внесения исправлений.

3) Найдите три ошибки в программе. Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки выпишите строку, в которой сделана ошибка и укажите, как исправить ошибку, т. е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

### Решение

Для решения был выбран код на языке Pascal.

1) Программа не выдаст никакой результат, так как запрашивается значение переменной A, но далее она нигде не фигурирует.

2) Ошибка 1: `maxDigit := N mod 10;`

Исправление 1: `maxDigit := -1;` (Следует проинициализировать её числом, меньшим нуля, например, -1, что будет означать, что пока в числе не найдена цифра, которая делится на 3).

3) Ошибка 2: `if maxDigit = 0 then writeln('Her')`

Исправление 2: `if maxDigit = -1 then writeln('Her')` (И так как мы условились



изначально проинициализировать maxDigit как -1, то именно это значение и стоит использовать для обнаружения отсутствия ответа)

4) Ошибка 3: readln(A);

Исправление 3: readln(N); (переменная A нигде более не используется, вся работа ведется с N).

25.

Задание на 8 баллов. Синтез логических выражений

Дана таблица истинности (представлена на рисунке). Необходимо синтезировать логическое выражение, соответствующее данной таблице. Последовательно выполнить все задания. В строке ввода расписать ответы на каждый пункт задания:

1. Выбрать наиболее простой способ синтеза.
2. Построить логическое выражение, соответствующее приведённой таблице истинности с помощью СДНФ или СКНФ.
3. Упростить данное логическое выражение с помощью законов и тождеств алгебры логики.

При вводе ответа использовать следующие обозначения логических связей (операций):

Отрицание: знак -

Дизъюнкция (логическое сложение): знак +

Конъюнкция (логическое умножение) знак можно опускать или использовать знак &.

Примеры:  $\bar{A} \wedge B$

$\neg AB$

$\bar{C} \wedge (A \vee B)$

$\neg C(A+B)$

$A \wedge B \wedge C \vee A \wedge \bar{B} \wedge \bar{C}$      $ABC+A-B-C$  или  $A\&B\&C+A-B-C$

A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

### Решение

1) Наиболее простым способом является выбор строк, где X=1.

2) 1 строка: X=1, тогда получаем значение переменных -A; -B; -C, соответственно X= (-A-B-C)

5 строка: X=1, тогда получаем значение переменных A; -B; -C, соответственно X= (A-B-C)

Так как функция X=1, значит составляем СДНФ X= (-A-B-C)+(A-B-C)

3) Далее с помощью законов и тождеств алгебры логики минимизируем логическое выражение X= (-A-B-C)+(A-B-C)= -B-C(-A+A)= -B-C



## **КРИТЕРИИ ОЦЕНИВАНИЯ ОТВЕТОВ ПОСТУПАЮЩИХ**

Оценка проводится по 100 – балльной шкале. Правильное решение заданий с 1-15 оценивается в 3 балла, правильное решение заданий с 16-18 оценивается в 4 балла, заданий с 19-21 оценивается в 5 балла, заданий с 22,23 оценивается в 6 баллов, правильное решение 24,25 задания оценивается в 8 баллов, для 22 - 25 заданий возможно выставление частичной оценки, если комиссия решит, что задание абитуриентом выполнено частично.

### **Особенности проведения вступительных испытаний для лиц с ограниченными возможностями здоровья и инвалидов**

1. Институт обеспечивает проведение вступительных испытаний для поступающих из числа лиц с ограниченными возможностями здоровья и (или) инвалидов (далее вместе - поступающие с ограниченными возможностями здоровья) с учетом особенностей их психофизического развития, их индивидуальных возможностей и состояния здоровья (далее - индивидуальные особенности).

2. В Институте должны быть созданы материально-технические условия, обеспечивающие возможность беспрепятственного доступа поступающих с ограниченными возможностями здоровья в аудитории, туалетные и другие помещения, а также их пребывания в указанных помещениях (в том числе наличие пандусов, подъемников, поручней, расширенных дверных проемов, лифтов; при отсутствии лифтов аудитория должна располагаться на первом этаже здания).

3. Вступительные испытания для поступающих с ограниченными возможностями здоровья проводятся в отдельной аудитории.

Число поступающих с ограниченными возможностями здоровья в одной аудитории не должно превышать 6 человек.

Допускается присутствие в аудитории во время сдачи вступительного испытания большего числа поступающих с ограниченными возможностями здоровья, а также проведение вступительных испытаний для поступающих с ограниченными возможностями здоровья в одной аудитории совместно с иными поступающими, если это не создает трудностей для поступающих при сдаче вступительного испытания.

Допускается присутствие в аудитории во время сдачи вступительного испытания ассистента из числа работников Института или привлеченных лиц, оказывающего поступающим с ограниченными возможностями здоровья необходимую техническую помощь с учетом их индивидуальных особенностей (занять рабочее место, передвигаться, прочитать и оформить задание, общаться с преподавателями, проводящими вступительное испытание).

4. Продолжительность вступительного испытания для поступающих с ограниченными возможностями здоровья увеличивается по решению Института, но не более чем на 1,5 часа.

5. Поступающим с ограниченными возможностями здоровья предоставляется в доступной для них форме информация о порядке проведения вступительных испытаний.

6. Поступающие с ограниченными возможностями здоровья могут в процессе сдачи вступительного испытания пользоваться техническими средствами, необходимыми им в связи с их индивидуальными особенностями.

### **СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ И ИНТЕРНЕТ-РЕСУРСОВ**

1. Поляков, К. Ю. Информатика: 10 класс: базовый и углублённый уровни. В 2 частях. Ч.1 : учебник / К. Ю. Поляков, Е. А. Еремин. — 6-е изд. — Москва : Просвещение, 2024. — 352 с.
2. Поляков, К. Ю. Информатика: 10 класс: базовый и углублённый уровни. В 2 частях. Ч.2 : учебник / К. Ю. Поляков, Е. А. Еремин. — 6-е изд. — Москва : Просвещение, 2024. — 352 с.
3. Поляков, К. Ю. Информатика: 11 класс: базовый и углублённый уровни. В 2 частях. Ч.1 : учебник / К. Ю. Поляков, Е. А. Еремин. — 6-е изд. — Москва : Просвещение, 2024. — 240 с.
4. Поляков, К. Ю. Информатика: 11 класс: базовый и углублённый уровни. В 2 частях. Ч.2 : учебник / К. Ю. Поляков, Е. А. Еремин. — 6-е изд. — Москва : Просвещение, 2024. — 304 с.
5. Кулабахина С. Ю., ЕГЭ Информатика 2025. 20 тренировочных вариантов. – М.: Легион, 2024.
6. Богомолова О. Б., ЕГЭ. Информатика. Новый полный справочник для подготовки к ЕГЭ. – М.: АСТ, 2023.
7. Крылов, С. С., Чуркина, Т. Е. ЕГЭ-2026. Информатика. Типовые экзаменационные варианты. 20 вариантов. – изд. Национальное образование, 2025.
8. СДАМ ГИА: РЕШУ ВПР, ОГЭ, ЕГЭ, ГВЭ, ЦТ, ЕНТ (URL: <https://sdamgia.ru/>)