

ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ

**ЕГЭ-2022**



Д.М. УШАКОВ

# ИНФОРМАТИКА

**10 ТРЕНИРОВОЧНЫХ ВАРИАНТОВ  
ЭКЗАМЕНАЦИОННЫХ РАБОТ  
ДЛЯ ПОДГОТОВКИ  
К ЕДИНОМУ  
ГОСУДАРСТВЕННОМУ ЭКЗАМЕНУ**

- ☒ **ДЛЯ ПОДГОТОВКИ К ЭКЗАМЕНУ  
В КОМПЬЮТЕРНОЙ ФОРМЕ**
- ☒ **ВАРИАНТ С ПОДРОБНЫМ  
РАЗБОРОМ РЕШЕНИЙ**



**ЕГЭ – ШКОЛЬНИКАМ  
И УЧИТЕЛЯМ**

**100  
БАЛЛОВ**

УДК 373:002  
ББК 32.81я721  
У93

**Ушаков, Денис Михайлович.**

У93      ЕГЭ-2022 : Информатика : 10 тренировочных вариантов экзаменационных работ для подготовки к единому государственному экзамену / Д.М. Ушаков. — Москва: АСТ, 2021. — 174, [2] с. — (ЕГЭ-2022. Это будет на экзамене).

ISBN 978-5-17-133166-5

Вниманию школьников и абитуриентов предлагается пособие для подготовки к ЕГЭ, которое содержит 10 тренировочных вариантов экзаменационных работ. Особенностью данного сборника является вариант с подробным разбором каждого задания. Представленные алгоритмы будут полезны при проверке и оценке своих навыков и умений в решении типовых экзаменационных заданий.

ЕГЭ по информатике, начиная с 2021 года, проводится в компьютерной форме. Каждый экзаменационный вариант в сборнике составлен в соответствии с требованиями единого государственного экзамена, включает задания разных типов и сложности по основным разделам курса информатики.

В конце пособия даны ответы для самопроверки на все задания.

Материал сборника может быть использован для планомерного повторения изученного материала и тренировки выполнения заданий различного типа при подготовке к единому государственному экзамену.

УДК 373:002  
ББК 32.81я721

ISBN 978-5-17-133166-5

© Ушаков Д.М., 2021  
© ООО «Издательство АСТ», 2021

# Содержание

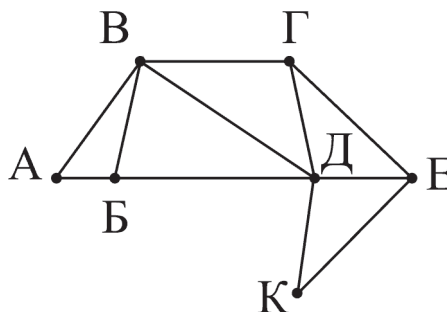
<b>Предисловие .....</b>	<b>4</b>
<b>Инструкция по выполнению работы .....</b>	<b>5</b>
<b>ВАРИАНТ С ПОДРОБНЫМ РАЗБОРОМ РЕШЕНИЙ .....</b>	<b>7</b>
<b>ВАРИАНТЫ ТРЕНИРОВОЧНЫХ РАБОТ</b>	
Вариант 1 .....	50
Вариант 2 .....	62
Вариант 3 .....	74
Вариант 4 .....	86
Вариант 5 .....	99
Вариант 6 .....	111
Вариант 7 .....	123
Вариант 8 .....	135
Вариант 9 .....	147
Вариант 10 .....	159
<b>Система оценивания экзаменационной работы .....</b>	<b>171</b>
<b>ОТВЕТЫ .....</b>	<b>172</b>

## ВАРИАНТ С ПОДРОБНЫМ РАЗБОРОМ РЕШЕНИЙ

1

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

	П1	П2	П3	П4	П5	П6	П7
П1		40	50				
П2	40		15		60		
П3	50	15		25	30		
П4			25		70	20	
П5		60	30	70		35	45
П6				20	35		55
П7					45	55	



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 длина дороги из пункта В в пункт Д. В ответе запишите целое число — так, как оно указано в таблице.

Ответ: \_\_\_\_\_.

### Решение.

Для каждой вершины графа посчитаем количество её соседей. Получим: А2, Б3, В4, Г3, Д5, Е3, К2. В данном случае видно, что вершины В и Д по этому параметру уникальные — других вершин с 4-мя и 5-ю соседями на графе нет. Найдём эти вершины в таблице. Для этого найдём строки, в которых, соответственно, 4 и 5 чисел. Это строки П3 и П5. На пересечении строки П3 и столбца П5 находится число 30.

Ответ: 30.

2

Вася заполнял таблицу истинности функции  $(w \wedge z) \vee \neg y \vee (\neg x \equiv \neg w)$ , но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных  $w, x, y, z$ :

				$(w \wedge z) \vee \neg y \vee (\neg x \equiv \neg w)$
1		0	0	0
	1		0	0
1	0	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.* Если бы функция была задана выражением  $\neg x \vee y$ , зависящим от двух переменных, а фрагмент таблицы имел бы вид:

		$\neg x \vee y$
0	1	0

то первому столбцу соответствовала бы переменная  $y$ , а второму столбцу — переменная  $x$ . В ответе следовало бы написать  $yx$ .

Ответ: \_\_\_\_\_.

### Решение.

Будем решать задачу путём последовательных умозаключений.

1. Данная функция дизъюнктивна, и при этом равна нулю в данной таблице. Это возможно только в случае, когда каждое её «слагаемое» равно нулю. То есть,  $\neg y = 0$ ,  $(w \wedge z) = 0$  и  $(\neg x \equiv \neg w) = 0$ . Чтобы выполнялось  $\neg y = 0$ , нужно, чтобы  $y = 1$ . Следовательно, в таблице должен быть столбец, состоящий из одних единиц. Единственный вариант для этого — 1-й столбец. Значит, в его 2-й строке стоит тоже 1.

2. Теперь рассмотрим требование  $(\neg x \equiv \neg w) = 0$ . Это возможно только тогда, когда  $x$  и  $w$  разные. Среди оставшихся 2-го, 3-го и 4-го столбцов для переменных  $x$  и  $w$  не подходит пара «2-й и 3-й», потому что в 3-й строке стоят оба нуля (а  $x$  и  $w$  должны быть разными). Также для  $x$  и  $w$  не подходит пара «3-й и 4-й», потому что в первой строке стоят оба нуля. Следовательно, для  $x$  и  $w$  остаётся только пара «2-й и 4-й». Значит, 3-й столбец — это переменная  $z$ .

3. Так как 2-й и 4-й столбцы соответствуют переменным  $x$  и  $w$ , а они должны быть разными, заполним в них пустые ячейки. В обоих случаях это должна быть единица, потому что соответствующая им цифра другого столбца равна нулю.

Рассмотрим полученную на данный момент таблицу. В ней нижними индексами помечены значения, которые были получены в результате соответствующих умозаключений.

$y_1$		$z_2$		$\neg y \vee (w \wedge z) \vee (\neg x \equiv \neg w)$
1	$1_3$	0	0	0
$1_1$	1		0	0
1	0	0	$1_3$	0

4. Чтобы отличить друг от друга переменные  $x$  и  $w$  (2-й и 4-й столбцы), нужно сделать вывод из оставшегося нерассмотренным выражения  $(w \wedge z) = 0$ . Для 1-й и 3-й строки таблицы это выражение выполняется. Необходимо заполнить оставшуюся ячейку.

В ней может стоять только цифра 1 (если поставить в эту ячейку 0, то получится две одинаковые строки таблицы истинности (1-я и 2-я), что невозможно по определению таблицы истинности). Получается, чтобы во 2-й строке выражение  $(w \wedge z) = 0$  было верным, должно быть  $w = 0$ . Среди 2-го и 4-го столбцов это верно только для 4-го столбца. Следовательно, 4-й столбец — это  $w$ , а 2-й столбец — это  $x$ .

**Ответ:**  $yxzw$ .

3

Ниже представлены два фрагмента таблиц из базы данных о жителях микрорайона. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных ID мужчины, ставшего отцом в наиболее старшем возрасте. При вычислении ответа учитывайте только информацию из приведённых фрагментов таблиц.

**Таблица 1**

ID	Фамилия_И.О.	Пол	Год_рождения
14	Дени К.Д.	Ж	2009
21	Ирта О.Б.	Ж	1988
26	Гано И.Е.	Ж	1962
34	Ивенко Д.Я.	М	1939
42	Бабенко А.Е.	М	1960
46	Ерава К.Д.	М	1957
53	Марченко Е.Е.	М	1928
57	Арне А.А.	М	2017
65	Лима С.Д.	М	1990
71	Камо Е.А.	М	1997
78	Исава Г.А.	Ж	1934
82	Сорт А.К.	Ж	1992
93	Викле И.С.	М	1913
	...		

**Таблица 2**

ID_Родителя	ID_Ребёнка
93	78
53	42
78	26
34	26
46	82
42	71
42	21
26	71
26	21
82	57
71	57
21	14
65	14
...	...

**Ответ:** \_\_\_\_\_.

### Решение.

В данном случае, задача не требует для своего решения построения графа родственных отношений. Так как вопрос только про отца, достаточно проанализировать отношения одного уровня и посмотреть на разницу в возрасте между отцом и ребёнком.



Будем анализировать, в основном, таблицу 2. Сначала выясним, в каких строках таблицы 2 родитель мужского пола. Это будут строчки:

ID_Родителя	ID_Ребёнка
93	78
53	42
34	26
46	82
42	71
42	21
71	57
65	14

Теперь для каждой этой строки выпишем год рождения родителя и ребёнка и посчитаем разницу между ними:

ID_Родителя	ID_Ребёнка	Разница в возрасте
93 (1913)	78 (1934)	21
53 (1928)	42 (1960)	32
34 (1939)	26 (1962)	23
46 (1957)	82 (1992)	35
42 (1960)	71 (1997)	37
42 (1960)	21 (1988)	28
71 (1997)	57 (2017)	20
65 (1990)	14 (2009)	19

Наибольшая разница (37) в строке 42–71.

**Ответ:** 42.

4


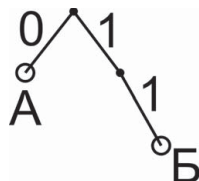
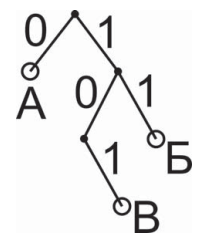
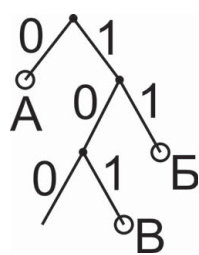
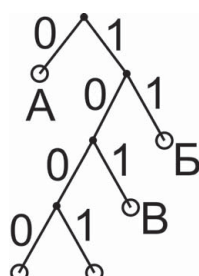
Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г, Д, решили использовать неравномерный двоичный код, удовлетворяющий условию, что никакое кодовое слово не является началом другого кодового слова. Это условие обеспечивает возможность однозначной расшифровки закодированных сообщений. Для букв А, Б, В использовали соответственно кодовые слова 0, 11, 101. Для двух оставшихся букв — Г и Д — кодовые слова неизвестны.

Укажите кратчайшее возможное кодовое слово для буквы Г, при котором код будет удовлетворять указанному условию. Если таких кодов несколько, укажите код с наименьшим числовым значением.

**Ответ:** \_\_\_\_\_.

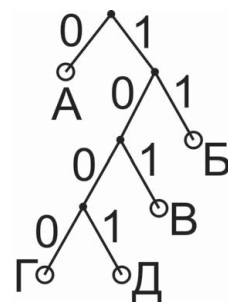
### Решение.

Будем строить «дерево декодирования» — двоичное дерево, при помощи которого обычно происходит декодирование неравномерных кодов, удовлетворяющих условию «никакое кодовое слово не является началом другого кодового слова». Также дерево декодирования очень удобно использовать при построении, а также при анализе неравномерных двоичных кодов.

<p>Будем строить двоичное дерево сверху вниз (сверху — корень, снизу — листья). Чтобы ничего не забыть, начнём строить дерево последовательно, дорисовывая в дерево листья для каждого по порядку кодового слова. Каждой цифре кодового слова соответствует ветка дерева декодирования. Для определённости будем рисовать ветви дерева для цифр 0 налево, а для цифр 1 — направо. Стартуем с буквы А. Её код — 0. То есть, от корня ветку нарисует налево. Узлы дерева отметим жирными точками, а листья (в данном случае, для буквы А) — кружками. Подпишем букву А рядом с кружком.</p>	
<p>Теперь сделаем такое же действие для буквы Б. Её кодовое слово — 11. От корня дерева (который уже нарисован) нарисует ветвь вправо (для 1) и от неё ещё одну ветвь вправо (для второй 1). Обозначим узел между ветвями (жирной точкой) и лист (кружком) для буквы Б.</p>	
<p>Дорисует дерево для буквы В. Её кодовое слово — 101. Первая ветвь — (1, вправо от корня) в дереве уже нарисована (когда мы рисовали ветви для буквы Б). От её конца чертим ветвь влево (для 0), а от её конца — ветвь вправо (для второй 1). Обозначим лист для буквы В.</p>	
<p>В полученном дереве ищем, куда бы ещё можно было пририсовать ветви. Нам нужно пририсовать ещё две ветви для букв Г и Д. Найдём узлы дерева, из которых выходит только одна ветвь. Такой узел находим только один. Путь к нему от корня — 10. В узле не хватает ветви влево, то есть, для 0. Получится ветвь 100. Следовательно, ветвь с листом для буквы Г или буквы Д можно пририсовать сюда.</p>	
<p>Если так сделать, то для второй буквы места уже не останется. Значит, в узле 100 не рисуем лист, а сделаем ещё один узел, из которого рисуем две ветви, налево и направо, для букв Г и Д.</p>	



У кодовых слов для букв Г и Д (1000 и 1001) получилась одинаковая длина. Так как для буквы Г нужно выбрать код с наименьшим числовым значением, выберем для буквы Г код 1000.



**Ответ:** 1000.

5

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: если  $N$  чётное, в конец числа (справа) дописывается 01, в противном случае справа дописывается 10.

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 81 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

**Ответ:** \_\_\_\_\_.

**Решение.**

Возьмём минимальное число, которое допустимо по условию («превышает 81»). Это число 82. Переведём его в двоичную систему счисления:

$$82 = 64 + 18 = 2^6 + 16 + 2 = 2^6 + 2^4 + 2^1 = 1010010_2.$$

Исследуем, может ли это число являться результатом работы указанного алгоритма. Так как в результате работы алгоритма к числу  $N$  добавляется ещё две двоичные цифры, отбросим от двоичного представления числа 82 две последние цифры. Получим  $10100_2$ . Применим к получившемуся числу действие 2) алгоритма. Число  $10100_2$  чётное (оканчивается на 0). Поэтому допишем 01 к числу:  $1010001_2$ . Сравним полученный результат с двоичным представлением числа 82 ( $1010010_2$ ). Так как этот результат оказался меньше числа 82, делаем вывод, что число 82 не подходит в качестве ответа. Также это означает, что текущий вариант числа  $N$  ( $10100_2$ ) также не подходит. Возьмём следующее число  $N$  (на 1 больше). Это число  $10101_2$ . Оно нечётное. Если применить к нему действие 2) алгоритма, получится число  $1010110_2$ . Сравним его с двоичным представлением числа 82. Оно больше. Получаем, что это число является результатом работы алгоритма, оно больше 81, и оно минимальное возможное. Переведём его в десятичную систему счисления.  $1010110_2$ .

**Ответ:** 86.

6

Определите, при каком наименьшем введённом значении переменной  $s$  программа выведет число 72.

Python	Паскаль
<pre> k = 0 s = int(input()) while s &lt; 205:     s = s + 10     k = k + 8 print(k) </pre>	<pre> var k, s : integer; begin     k := 0;     readln(s);     while s &lt; 205 do         begin             s := s + 10;             k := k + 8         end;     write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include &lt;iostream&gt; using namespace std; int main() {     int k, s;     k = 0;     cin &gt;&gt; s;     while (s &lt; 205) {         s = s + 10;         k = k + 8;     }     cout &lt;&lt; k &lt;&lt; endl;     return 0; } </pre>	<pre> алг нач     цел k, s     k := 0     ввод s     нц пока s &lt; 205         s := s + 10         k := k + 8     кц     вывод k кон </pre>

Ответ: \_\_\_\_\_.

### Решение.

Анализируем программу. В ней в цикле значение переменной  $k$  становится равным 72. Начальное значение переменной равно нулю. На каждом шаге цикла к ней прибавляется 8. Найдём количество раз  $n$ , которое выполнился цикл:  $0 + 8n = 72 \Rightarrow n = 9$ . Теперь рассмотрим переменную  $s$ . У неё перед циклом было какое-то начальное значение. На каждом шаге цикла к этой переменной прибавлялось 10, а цикл при этом выполнялся 9 раз. Значит, к начальному значению переменной  $s$  было прибавлено число  $10 \cdot 9 = 90$ , и цикл в результате закончился. То есть, перестало выполняться условие  $s < 205$ . Следовательно, после окончания цикла  $s \geq 205$ .

Получаем неравенство относительно начального значения переменной  $s$ :

$$s + 90 \geq 205 \Rightarrow s \geq 115.$$

Наименьшее значение переменной  $s$ , при которой это возможно — 115.

Ответ: 115.

7

Для хранения произвольного растрового изображения размером  $256 \times 512$  пикселей отведено 80 Кбайт памяти без учёта размера заголовка файла. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Какое максимальное количество цветов можно использовать в изображении?

Ответ: \_\_\_\_\_.

**Решение.**

Воспользуемся формулой:

$$V = H \cdot W \cdot i,$$

где  $V$  — размер файла с изображением в битах,  
 $H$  и  $W$  — высота и ширина изображения в пикселях,  
 $i$  — количество бит в одном пикселе.

Подставим в эту формулу известные величины, переведя Кбайты в биты.

Получим:  $80 \cdot 2^{13} = 256 \cdot 512 \cdot i$ . Отсюда,  $i = 5$ .

По формуле:

$$2^{\text{число\_бит}} = \text{число\_цветов}$$

найдем количество цветов:  $2^5$ .

Ответ: 32.

8

Степан составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Степан использует четырёхбуквенные слова, в которых могут быть только буквы С, Т, Е, П, А, Н, причём буква Т появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Степан?

Ответ: \_\_\_\_\_.

**Решение.**

Сначала проанализируем, сколькими способами можно поставить в слово букву Т.

На любую из четырёх позиций четырёхбуквенного слова, т. е. 4. На оставшиеся 3 позиции можно поставить любую из оставшихся пяти букв — это  $5^3$ .

Так как нужно сделать и то, и другое, перемножаем эти значения:

$$4 \cdot 5^3 = 4 \cdot 125 = 500.$$

Ответ: 500.



*Задание выполняется с использованием прилагаемых файлов.*

<https://dmushakov.ru/files/EGE-2022.zip> папка 09 файл 09А

9

Откройте файл электронной таблицы **09А**, содержащей вещественные числа — результаты ежечасного измерения скорости ветра на протяжении трёх месяцев. Найдите разность между средним арифметическим значением скорости ветра и её минимальным значением.

В ответе запишите только целую часть получившегося числа, с округлением по правилам математики.

Ответ: \_\_\_\_\_.

### Решение.

Откроем файл **09А.xlsx** из папки **09**. Найдём среднее арифметическое значение скорости ветра. Для этого, например, в ячейке **Z1** запишем формулу, вычисляющую среднее арифметическое: `=СРЗНАЧ(`. Для бесплатного Office, нужно будет написать `=AVERAGE(`. Курсор будет стоять после открывающей скобки. В качестве аргумента функции `СРЗНАЧ` укажем диапазон всех значений скорости ветра — это прямоугольный блок ячеек от **B2** до **Y93**. Напишем номера этих ячеек через двоеточие или выделим их мышью, закроем круглую скобку и нажмём клавишу `[Enter]`.

Электронная таблица выдаст нам значение 12,2.

Теперь аналогично найдём минимальное значение для этого диапазона. Воспользуемся формулой `=МИН(` для MS Excel или `=MIN(` для бесплатного Office. Напишем её, например, в ячейке **Z2**. Не забудем закрыть круглую скобку и нажать на клавиатуре `[Enter]`. Получим результат 1,2.

В соседней ячейке (например, **Z3**) посчитаем разность между ними: `=Z1-Z2`.

Получим 11.

Ответ: 11.



*Задание выполняется с использованием прилагаемых файлов.*

<https://dmushakov.ru/files/EGE-2022.zip> папка 10 файл А.П. Чехов. Рассказы.docx

10

С помощью текстового редактора определите, сколько раз, не считая сносок, встречается слово «мой» или «Мой» в тексте А.П. Чехова «Рассказы». Другие формы слова «мой», такие как «мои», «моими» и т. д., учитывать не следует. В ответе укажите только число.

Ответ: \_\_\_\_\_.

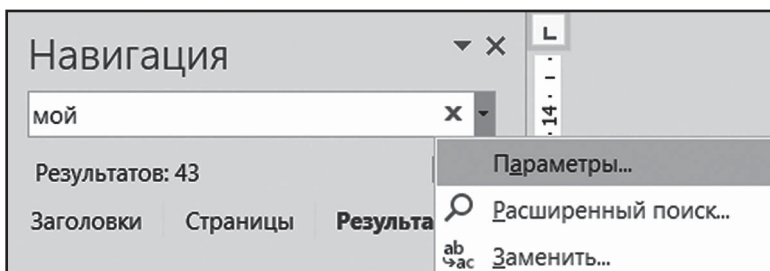
### Решение.

Откроем нужный файл в тестовом редакторе MS Word. Перейдём в режим поиска текста, нажав на клавиатуре `[Ctrl]+[F]` или щёлкнув по кнопке «Найти» панели «Глав-

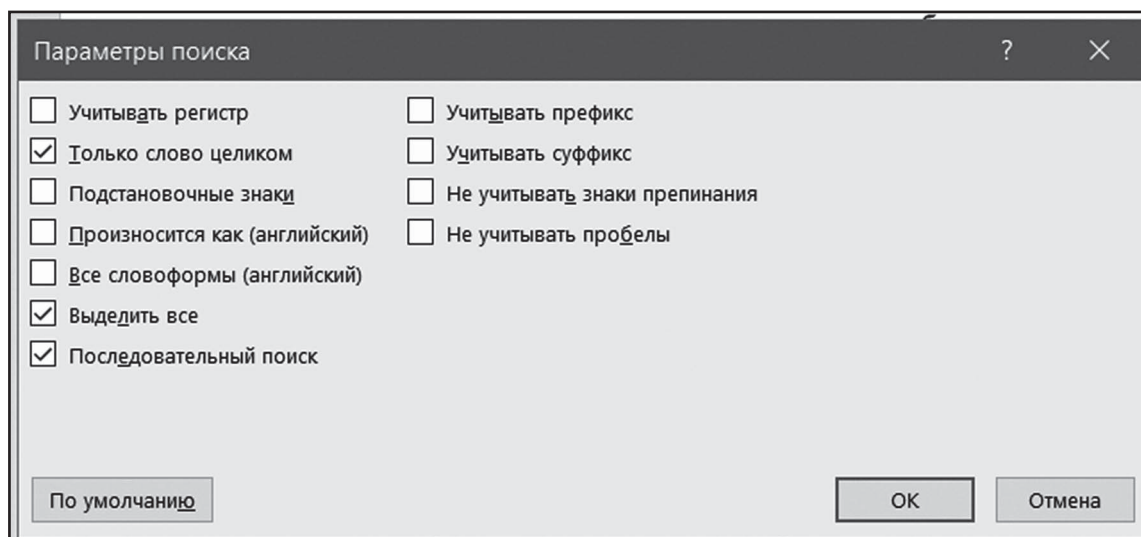
ная». Слева появится дополнительное окно-«докер» с названием «Навигация». Курсор клавиатуры при этом сразу будет стоять и мигать в поле ввода текста, который мы собираемся искать в тексте. Убедимся сначала, что MS Word ищет это слово с правильными параметрами.

Нас будут интересовать два параметра. Во-первых, что MS Word ищет это слова без учёта регистра. В этом случае будут найдены не только слова «мой», но и слова «Мой». Во-вторых, что MS Word не ищет при этом сочетание букв «мой» как часть другого слова, например, «самой».

Для этого щёлкнем по стрелке вниз справа от поля ввода и выберем в ниспадающем списке пункт «Параметры».



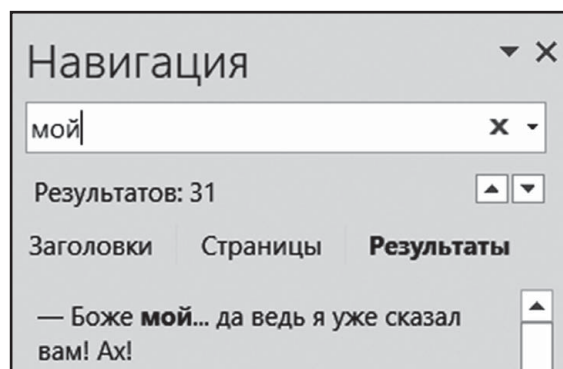
В появившемся окне должна быть не установлена «галочка» для параметра «Учитывать регистр» и при этом установлена «галочка» для параметра «Только слово целиком».



Подтвердим выбор щелчком по кнопке «ОК».

Теперь щёлкните мышью в поле ввода и введите в него требуемое слово «мой». При необходимости нажмите после этого кнопку [Enter]. Через несколько секунд под строкой поиска MS Word выведет количество найденных слов (31).

Ниже можно просмотреть фрагменты текста, в которых MS Word нашёл требуемый текст, и убедиться, что не найдено ничего лишнего.



**Ответ: 31.**

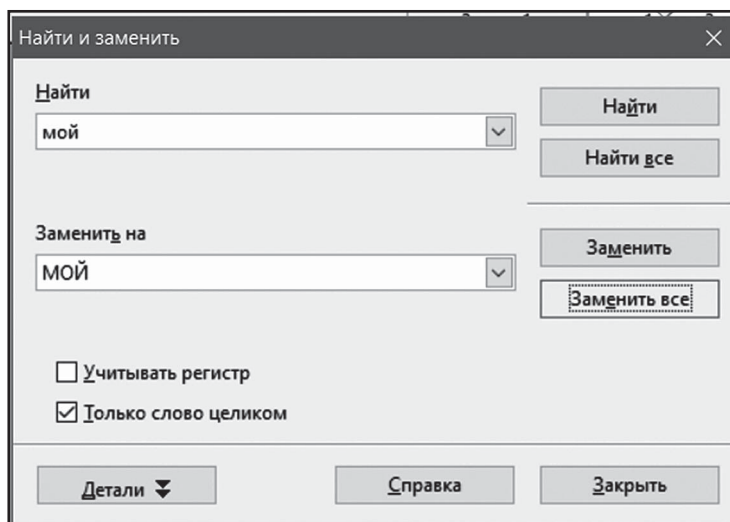
*Замечание.*

Если вы используете бесплатный Office, следует действовать чуть хитрее.

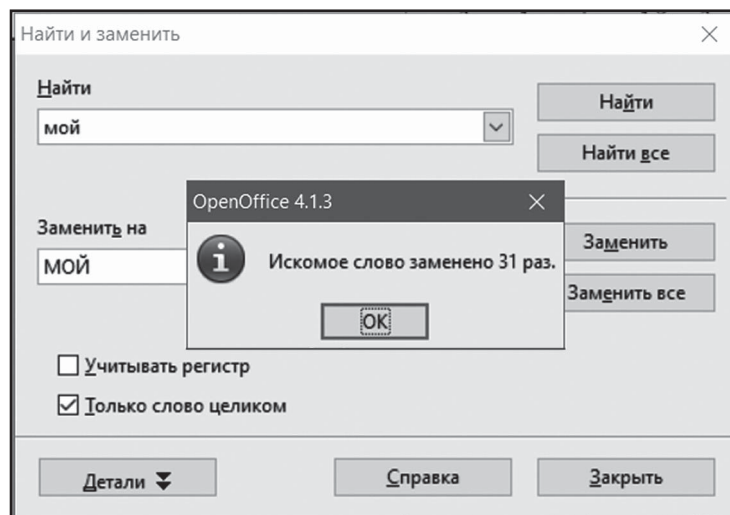
Рассмотрим решение в OpenOffice Writer. Откроем нужный файл и перейдём в режим поиска. Для этого нажмём на клавиатуре [Ctrl]+[F]. В появившемся окне «Найти и заменить» впишем в поле «Найти» требуемый текст «мой» и убедимся, что «галочка» для «Учитывать регистр» **не установлена**, а «галочка» для «Только слово целиком» **установлена**.

Чтобы узнать, сколько таких слов имеется в тексте, придётся либо вручную щёлкать по кнопке «Найти», либо воспользоваться действием «Заменить все».

Введём в поле «Заменить на» слово «МОЙ» и щёлкнем по кнопке «Заменить все».



OpenOffice Writer покажет нам, сколько он сделал замен.



«Искомое слово заменено 31 раз».

**Ответ: 31.**



11

При регистрации в компьютерной системе каждому пользователю выдается пароль, состоящий из 15 символов и содержащий только символы из 12-символьного набора: А, В, С, D, E, F, G, H, K, L, M, N. 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 отведено 12 байт на одного пользователя.

Определите объём памяти (в байтах), необходимый для хранения сведений о 30 пользователях. В ответе запишите только целое число — количество байт.

Ответ: \_\_\_\_\_.

### Решение.

Сначала воспользуемся **формулой Хартли** ( $2^i \geq N$ ) для вычисления количества информации, которое требуется для записи одного символа пароля. Всего возможных символов в пароле 12, поэтому из неравенства  $2^i \geq 12$  получаем, что для записи одного символа пароля требуется 4 бита. В пароле 15 символов, поэтому всего для пароля требуется  $15 \cdot 4 = 60$  бит. Теперь это нужно перевести в байты: делим 60 на 8, получаем 7,5. Следовательно, для пароля требуется 8 целых байт.

Ещё для каждого пользователя хранится 12 байт дополнительной информации. Значит, на одного пользователя вместе с паролем хранится  $8 + 12 = 20$  байт. Так как пользователей 30, умножаем одно на другое. 20 Байт  $\cdot$  30 пользователей = 600 Байт.

Ответ: 600.

12

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

**А) заменить ( $v, w$ ).**

Эта команда заменяет в строке первое слева вхождение цепочки  $v$  на цепочку  $w$ . Например, выполнение команды **заменить (222, 34)** преобразует строку 77222277 в строку 7734277.

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).

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 83 идущих подряд цифр 7? В ответе запишите полученную строку.

**НАЧАЛО**  
**ПОКА** нашлось (77777) **ИЛИ** нашлось (222)  
     **ЕСЛИ** нашлось (77777)  
         **ТО** заменить (77777, 22)  
         **ИНАЧЕ** заменить (222, 2)  
     **КОНЕЦ ЕСЛИ**  
**КОНЕЦ ПОКА**  
**КОНЕЦ**

Ответ: \_\_\_\_\_.

### Решение.

Удобнее всего эту задачу решать программным способом. Напишем подпрограммы **заменить** и **нашлось**, которые соответствующим образом обрабатывают строку, а потом перепишем программу с приведённого псевдоязыка на используемый язык программирования.

#### Питон

```
def nash(v):
    global s
    return s.find(v) >= 0
def zam(v,w):
    global s
    if nash(v):
        k = s.find(v)
        s = s.replace(v,w,1)
s = "7"*83
while nash('77777') or nash('222'):
    if nash('77777'):
        zam('77777', '22')
    else:
        zam('222', '2')
print(s)
```

## Паскаль

```

var s : string;
function nash(v : string): boolean;
begin
    nash := (pos(v,s)<>0)
end;
procedure zam(v,w : string);
var k : integer;
begin
    if nash(v) then
    begin
        k := pos(v,s);
        delete(s,k,length(v));
        insert(w,s,k)
    end;
end;
var i : integer;
begin
    s := '';
    for i:=1 to 83 do
        s := s + '7';
        while nash('77777') or
            nash('222') do
        begin
            if nash('77777') then
                zam('77777','22')
            else
                zam('222','2');
            writeln(s)
        end;
    end.

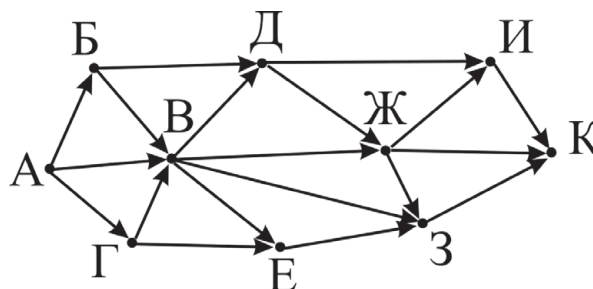
```

Запустив эту программу, получим в последней строчке ответ: 22777.

Ответ: 22777.

13

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



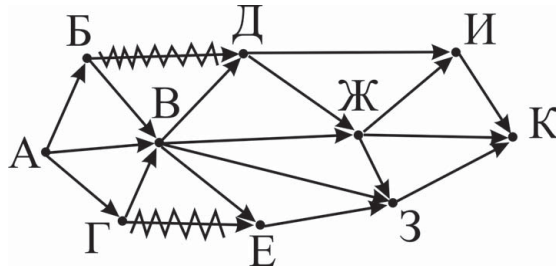
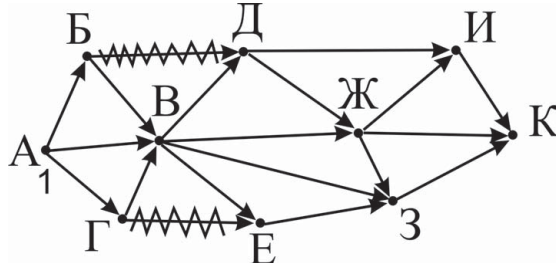
Ответ: \_\_\_\_\_.

### Решение.

Будем использовать для решения быстрый и эффективный способ решения под названием **динамическое программирование**.

Идея метода динамического программирования состоит в том, чтобы постепенно, шаг за шагом, получать частичные результаты для каждой рассматриваемой вершины. В результате работы метода на последнем шаге мы получим ответ для искомой вершины.

В данном случае интересующий нас результат — это сколькими различными способами можно «доехать» из стартового города до текущей вершины (города).

<p><b>1.</b> Так как в этой задаче нас спрашивают определить количество путей, проходящих через город В, сразу вычеркнем на схеме те стрелки, которые «обходят» город В.</p> <p>В данном случае, это будут стрелки <math>B \rightarrow D</math> и <math>\Gamma \rightarrow E</math>. Вычеркнем их на схеме.</p>	
<p><b>2.</b> Будем последовательно искать, сколькими различными способами можно добраться из города А до каждого из городов, нарисованных на схеме.</p> <p>Из города А до самого себя можно доехать единственным способом — никуда не ездя.</p> <p>Пометим вершину А числом 1 (количество способов, которым можно добраться до данной вершины).</p>	
<p><b>3.</b> Теперь постараемся пометить какую-нибудь ещё вершину. Это можно сделать только для тех вершин, про которые известно нужное количество для всех вершин, входящих в данную вершину. На данном этапе мы знаем нужное количество только для одной вершины — А.</p> <p>Будем искать вершины, в которые входит только одна стрелка дороги — из вершины А. Таких городов на данной схеме два — В и Г. Пометим их числами 1. Это количество способов добраться из города А в города В и Г.</p>	