

**ДЕМОНСТРАЦИОННЫЙ ВАРИАНТ ПО ПРЕДМЕТУ  
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

1. Позиционные системы счисления – 5
2. Построение таблиц истинности логических выражений – 7
3. Построение логического выражения по логической схеме – 7
4. Графы. Поиск путей – 6
5. Анализ информационных моделей - 6
6. Поиск информации в базе данных – 4
7. Выполнение и анализ простых алгоритмов – 7
8. Компьютерные сети – 4
9. Комбинаторика – 4
10. Адресация в электронных таблицах – 4
11. Количество информации – 5
12. Кодирование - 5
13. Рекурсивные алгоритмы – 5
14. Запросы для поисковых систем – 7
15. Программирование - 8
16. Программирование - 8
17. Программирование - 8

Итого – 100 баллов

**Ответом к заданию может быть:**

1. целое число, записывается без пробелов: 11;
2. дробное число, записывается через запятую без пробелов: 3,14;
3. последовательность цифр, записанных через точку, без пробелов:  
255.255.255.255
4. последовательность цифр или букв без пробелов: 30567, A, AB, 1AF4, A73,  
AEF, хуz, xXyZ.

## I. Системы счисления

### Задание 1

Решите уравнение и определите систему счисления  $x$ .

В ответе запишите только число в десятичной системе счисления

$$157_8 - 13_x = 64_{16}$$

**Ответ: 8**

### Решение:

Переведем в десятичную систему счисления числа  $157_8$  и  $64_{16}$ .

$$157_8 = 111_{10}$$

$$64_{16} = 100_{10}$$

Представим  $13_x$  в системе счисления

$$13_x = 1 \cdot x^1 + 3 \cdot x^0 = x + 3$$

Составим линейное уравнение в десятичной системе счисления, найдем  $x$ .

$$111 - (x + 3) = 100$$

$$x = 8$$

## II. Построение таблиц истинности логических выражений

### Задание

Логическая функция  $F$  задаётся выражением:

$$\overline{(x \rightarrow y \wedge \bar{z})} \equiv (z \wedge \bar{x})$$

Составьте таблицу истинности. В ответе запишите только число.

Определите количество строк, при которых функция  $F$  равна 0.

**Ответ: 5**

### Решение:

Составим таблицу истинности.

Таблица истинности – это таблица, определяющая значение сложного высказывания при всех возможных значениях простых высказываний.

Таблица истинности инверсии «НЕ» (знак  $\bar{\quad}$ ) – отрицание

|   |   |
|---|---|
| x | F |
| 0 | 1 |

|   |   |
|---|---|
| 1 | 0 |
|---|---|

Таблица истинности логического умножения «И» (знак  $\wedge$ ) – конъюнкция

| x | y | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Таблица истинности логического сложения «ИЛИ» (знак  $\vee$ ) – дизъюнкция

| x | y | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Таблица истинности импликации (знак  $\rightarrow$ ) – следование (ЕСЛИ – ТО)

| x | y | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Таблица истинности эквиваленции (знак  $\equiv$ ) – логическая равнозначность

| x | y | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Таблица истинности исключающее «ИЛИ» (знак  $\oplus$ )

| x | y | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Приоритет операций :

- 1) Выражение в скобках
- 2) Инверсия «НЕ»
- 3) Конъюнкция «И»
- 4) Дизъюнкция «ИЛИ»
- 5) Импликация
- 6) Эквивалентность
- 7) Исключающее «ИЛИ»

Построим таблицу истинности функции

$$F = (x \rightarrow y \wedge \bar{z}) \equiv (z \wedge \bar{x})$$

| x | y | z | $\bar{z}$ | $y \wedge \bar{z}$ | $x \rightarrow y \wedge \bar{z}$ | $\bar{x}$ | $z \wedge \bar{x}$ | $(x \rightarrow y \wedge \bar{z}) \equiv (z \wedge \bar{x})$ | F |
|---|---|---|-----------|--------------------|----------------------------------|-----------|--------------------|--|---|
| 0 | 0 | 0 | 1         | 0                  | 1                                | 1         | 0                  | 0  | 1 |
| 0 | 0 | 1 | 0         | 0                  | 1                                | 1         | 1                  | 1  | 0 |
| 0 | 1 | 0 | 1         | 1                  | 1                                | 1         | 0                  | 0  | 1 |
| 0 | 1 | 1 | 0         | 0                  | 1                                | 1         | 1                  | 1  | 0 |
| 1 | 0 | 0 | 1         | 0                  | 0                                | 0         | 0                  | 1  | 0 |
| 1 | 0 | 1 | 0         | 0                  | 0                                | 0         | 0                  | 1  | 0 |
| 1 | 1 | 0 | 1         | 1                  | 1                                | 0         | 0                  | 0  | 1 |
| 1 | 1 | 1 | 0         | 0                  | 0                                | 0         | 0                  | 1  | 0 |

Количество строк, при которых функция F равна 0 равно 5.

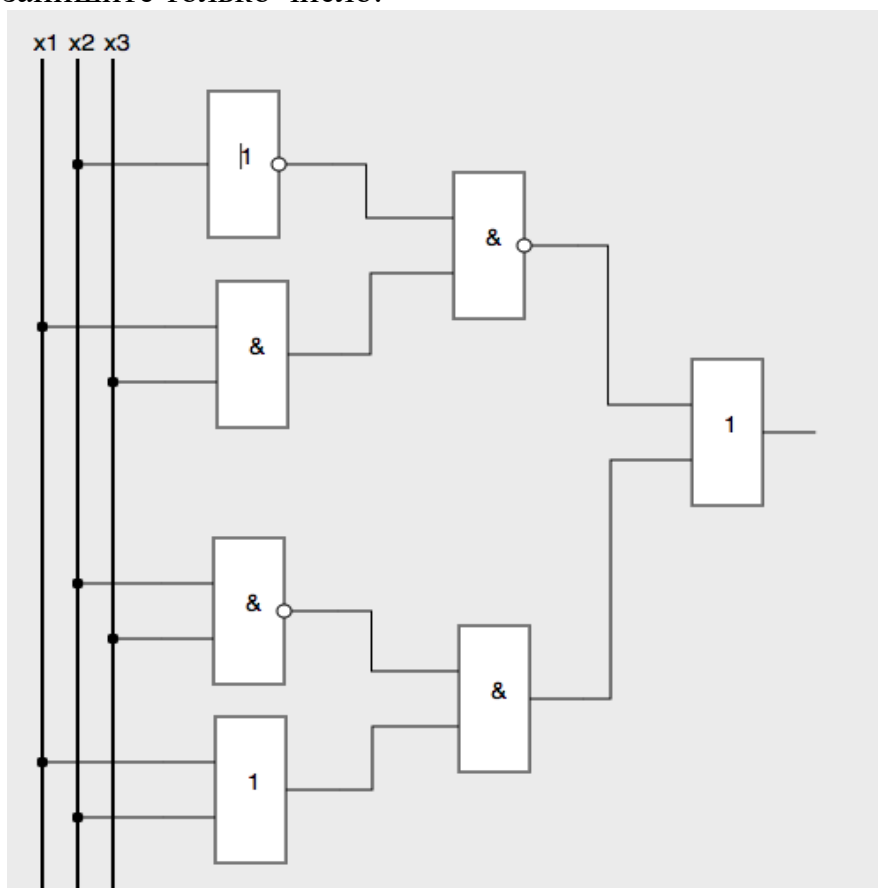
### Ш. Построение логического выражения по логической схеме

#### Задание

Построение логического выражения по логической схеме и составление таблицы истинности.

Определить количество строк, при которых функция F равна 0.

В ответе запишите только число.

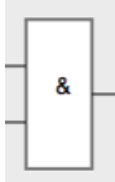


**Ответ: 0**

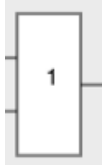
**Решение:**


Логические элементы, для создания логической схемы

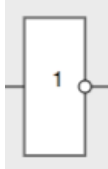
Конъюнкция – логическое «ИЛИ», знак на схеме ( & )



Дизъюнкция – логическое «И», знак на схеме ( 1 )



Инверсия – логическое отрицание «НЕ» , знак на схеме (  )



Составим таблицу истинности.

Таблица истинности – это таблица, определяющая значение сложного высказывания при всех возможных значениях простых высказываний.

Таблица истинности инверсии «НЕ» (знак  $\neg$ ) – отрицание

| x | F |
|---|---|
| 0 | 1 |
| 1 | 0 |

Таблица истинности логического умножения «ИЛИ»(знак  $\wedge$ )– конъюнкция

| x | y | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Таблица истинности логического сложения «И»(знак  $\vee$ )– дизъюнкция

| x | y | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Приоритет операций :

- 1) Выражение в скобках
- 2) Инверсия «НЕ»
- 3) Конъюнкция «И»
- 4) Дизъюнкция «ИЛИ»

Построим таблицу истинности для функции

$$F = \overline{x_2} \wedge (x_1 \wedge x_3) \vee \overline{x_2} \wedge x_3 \wedge (x_1 \vee x_2)$$

| x1 | x2 | x3 | $x_1 \wedge x_3$ | $\overline{x_2}$ | $\overline{x_2 \wedge (x_1 \wedge x_3)}$ | $\overline{x_2} \wedge x_3$ | $x_1 \vee x_2$ | $(x_2 \wedge x_3) \wedge (x_1 \vee x_2)$ | $\overline{x_2} \wedge x_3 \wedge (x_1 \vee x_2)$ | F |
|----|----|----|------------------|------------------|--|-----------------------------|----------------|--|---|---|
| 0  | 0  | 0  | 0                | 1                | 1  | 1                           |                | 0  | 0   | 1 |
| 0  | 0  | 1  | 0                | 1                | 1  | 1                           |                | 0  | 0   | 1 |
| 0  | 1  | 0  | 0                | 0                | 1  | 1                           |                | 1  | 1   | 1 |
| 0  | 1  | 1  | 0                | 0                | 1  | 0                           |                | 0  | 0   | 1 |
| 1  | 0  | 0  | 0                | 1                | 1  | 1                           |                | 1  | 1   | 1 |
| 1  | 0  | 1  | 1                | 1                | 0  | 1                           |                | 1  | 1   | 1 |
| 1  | 1  | 0  | 0                | 0                | 1  | 1                           |                | 1  | 1   | 1 |
| 1  | 1  | 1  | 1                | 0                | 1  | 0                           |                | 1  | 0   | 1 |

Вектор функция F: 11111111

Количество строк, при которых функция F равна 0 равно 0.

#### IV. Графы. Поиск путей

##### Задание

Сколько существует различных путей из пункта А в пункт М, проходящих через пункт Л?

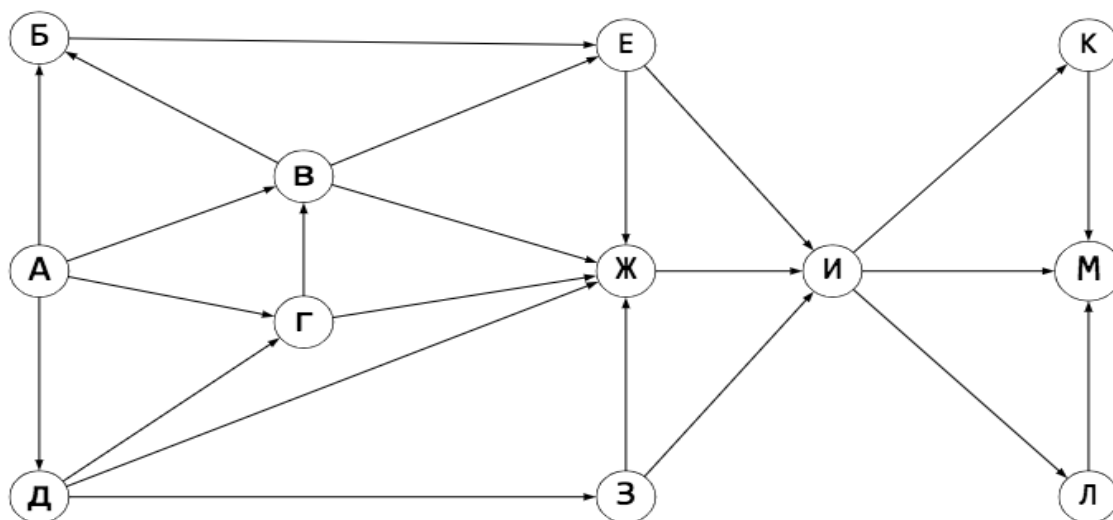


Схема дорог, представленная на рисунке, связывающих пункты А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

**Ответ: 19**

**Решение:**

Количество путей до пункта X - это количество путей добраться в любой из тех пунктов, из которых есть дорога в X. Если путь не должен проходить через какой-то пункт, нужно просто не учитывать этот пункт при подсчёте сумм. Если пункт должен лежать на пути, тогда для пунктов, в которые из нужного пункта идут дороги, в суммах нужно брать только этот пункт.

Подсчитаем последовательно количество путей до каждого из пунктов:

- 1)  $A = 1$
- 2)  $B = A = 1$
- 3)  $D = A = 1$
- 4)  $\Gamma = A + D = 2$
- 5)  $V = B + \Gamma + A = 4$
- 6)  $E = V + B = 5$
- 7)  $З = D = 1$
- 8)  $Ж = V + \Gamma + E + Д + З = 13$
- 9)  $И = Ж + E + З = 19$
- 10)  $Л = И = 19$
- 11)  $К = И = 19$
- 12)  $М = Л = 19$  (Пункты К и И не учитываем, поскольку путь должен проходить через Л)

Ответ: 19

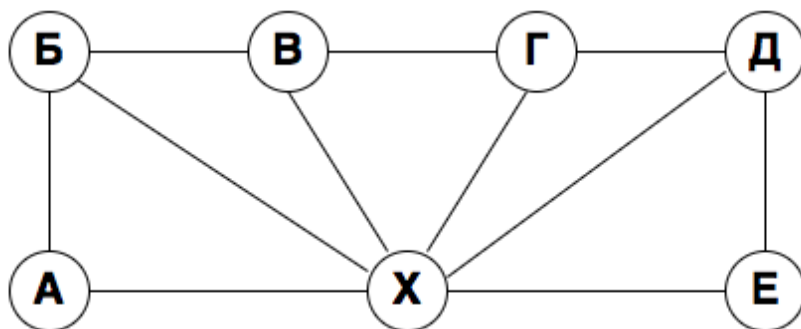
## V. Анализ информационных моделей

### Задание

На рисунке схема дорог N-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 сумму протяжённостей дорог из пункта Б в пункт В и из пункта Г в пункт Д.

В ответе запишите целое число.



|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
|    | П1 | П2 | П3 | П4 | П5 | П6 | П7 |
| П1 | ■  | 3  |    |    | 4  |    |    |
| П2 | 3  | ■  |    |    | 12 | 13 |    |
| П3 |    |    | ■  | 10 | 11 |    |    |
| П4 |    |    | 10 | ■  | 9  |    | 7  |
| П5 | 4  | 12 | 11 | 9  | ■  | 8  | 6  |
| П6 |    | 13 |    |    | 8  | ■  | 5  |
| П7 |    |    |    | 7  | 6  | 5  | ■  |

**Ответ: 20**

**Решение:**

Х - единственная вершина шестой степени, значит, Х соответствует П5. Вершины А и Е - единственные вершины степени 2, тогда они могут соответствовать П1 и П3. Вершины Б и Д связаны с вершинами А и Е, тогда из таблицы получаем, что они могут соответствовать П2 и П4. Тогда В и Г могут соответствовать П6 и П7.

Точное соответствие букв пунктам не важно.

Протяженность из пункта Б в пункт В равна 13.

Протяженность из пункта Г в пункт Д равна 7.

Сумма протяженностей дорог из пункта Б в пункт В и из пункта Г в пункт Д равна  $13 + 7 = 20$

## VI. Поиск информации в базе данных

**Задание**

Дан фрагмент базы данных в котором представлены сведения о родственных отношениях. На основе приведенных данных определите фамилию и инициалы внука Петровой С.М.

| ID  | Фамилия И.О.  | Пол |
|-----|---------------|-----|
| 25  | Жвания К.Г.   | ж   |
| 49  | Черняк А.П.   | м   |
| 62  | Ильченко С.И. | ж   |
| 76  | Ильченко Т.В. | ж   |
| 82  | Петрова С.М.  | ж   |
| 96  | Басис В.В.    | ж   |
| 102 | Ильченко В.И. | м   |
| 123 | Павлыш Н.П.   | ж   |
| 134 | Черняк П.Р.   | м   |

| ID Родителя | ID Ребенка |
|-------------|------------|
| 25          | 134        |
| 76          | 49         |
| 76          | 123        |
| 82          | 76         |
| 82          | 96         |
| 102         | 76         |
| 102         | 96         |
| 134         | 49         |
| 134         | 123        |

Выбери один из вариантов ответа:

- 1) Басис В.В.
- 2) Черняк А.П.
- 3) Павлыш Н.П.
- 4) Ильченко С.И.

**Ответ: 2**

**Решение:**

Находим ID Петровой С.М. – это ID 82. Это бабушка, далее определяем ее детей.

| ID | Фамилия И.О. |
|----|--------------|
| 82 | Петрова С.М. |

Определяем в таблице ID Ребенка с ID Родителя равное 82 – это ID Ребенка 76 и 96. Они являются родителями для следующего поколения.

| ID Родителя | ID Ребенка |
|-------------|------------|
| 82          | 76         |
| 82          | 96         |

Соответственно у ID Родителя равное 82 находим ID Ребенка – это ID Ребенка 49 и 123.

| ID Родителя | ID Ребенка |
|-------------|------------|
| 76          | 49         |
| 76          | 123        |
| 96          | -          |

ID Родителя с номером 76 соответствует ID Ребенка с номерами 49 и 123. Под номером 123 – пол “ж”. Следовательно, только у ID Родителя 49 пол «М» - это Черняк А.П. , внук Петровой С.М.  
Внук Петровой С.М.– Черняк А.П.

| ID Родителя | Фамилия И.О. | пол |
|-------------|--------------|-----|
| 49          | Черняк А.П.  | м   |
| 123         | Павлыш Н.П.  | ж   |

## VII. Выполнение и анализ простых алгоритмов

### Задание

На вход алгоритма подается натуральное число  $N$ . Алгоритм строит по нему новое число  $X$  следующим образом:

1. Строится двоичная запись числа  $N$ .
2. К этой записи дописывается 1.
3. Далее к этой записи дописываются бит четности по следующему правилу:
  - если в двоичном коде полученного числа четное количество единиц, то к двоичной записи числа в конце дописывается 0.
  - если в двоичном коде полученного числа нечетное количество единиц, то к двоичной записи числа в конце дописывается 1.
4. К полученному числу дописывается еще один бит четности.

Полученная запись является двоичной записью искомого числа  $X$ , в ней на три разряда больше, чем в записи исходного числа  $N$  и это число является двоичной записью искомого числа  $X$ .

Какое минимальное число  $X$ , большее 168, может быть получено в результате работы этого алгоритма ?

**Ответ: 172**

### Решение:

Есть входное число  $N$ . К нему дописывается 1. Далее, если количество единиц в этом числе четное, соответственно сумма цифр четная, то дописывается 0. Если количество единиц нечетное, то в конце дописывается 1, следовательно сумма становится четной. Если мы дописали 1, то сумма будет четной и по пункту 4 мы дописываем 0. Если же мы дописали 0, то сумма числа не изменилась и мы тоже дописываем 0.

Отсюда следует, число в двоичной системе счисления заканчивается на 100 или 110.

Нужно найти такое число  $X$ , больше, чем 168, и это число в двоичной системе счисления заканчивается на 100 или 110.

Рассмотрим ближайшие числа к 168:

1. Возьмем число  $169_{10} = 10\ 101\ 001_2$   
Это число заканчивается на 001 – не подходит.
2. Возьмем число  $170_{10} = 10\ 101\ 010_2$   
Это число заканчивается на 010 – не подходит.
3. Возьмем число  $171_{10} = 10\ 101\ 011_2$   
Это число заканчивается на 011 – не подходит.
4. Возьмем число  $172_{10} = 10\ 101\ 100_2$   
Это число заканчивается на 100 – подходит.

### VIII. Компьютерные сети

#### Задание

Маска подсети 255.255.224.0 и IP-адрес компьютера в сети 10.8.248.131.

Определить адрес сети.

Выберите один из вариантов ответа:

- 1) 10.8.224.0
- 2) 255.255.248.131
- 3) 10.255.248.0
- 4) 255.8.224.131

**Ответ: 1**

#### Решение:

**IP-адрес** - это уникальный адрес, который идентифицирует устройство в Интернете, работающей по протоколу IP. В версии протокола IPv4 IP-адрес имеет длину 4 байта.

**Маска подсети** – битовая маска, позволяющая разделить IP-адрес на адрес подсети и адрес узла (хоста, компьютера, устройства) внутри этой подсети.

**IP-адрес** компьютера и маска подсети состоит из четырех октетов, разделенных точками.

Информация в октете записывается в виде целого числа от 0 до 255:

Маска подсети 255.255.224.0

IP-адрес компьютера в сети 10.8.248.131

1. Двоичный код IP-адреса  
 $10.8.248.131 \rightarrow 0000\ 1010.\ 0000\ 1000.\ 1111\ 1000.\ 1000\ 0011$
2. Двоичный код маски  
 $255.255.224.0 \rightarrow 1111\ 1111.\ 1111\ 1111.\ 1110\ 0000.\ 0000\ 0000$
3. Операция поразрядной конъюнкции (логическое умножение) для маски и IP-адреса:  
 $0000\ 1010.\ 0000\ 1000.\ \mathbf{1111\ 1000}.\ \mathbf{1000\ 0011}$

1111 1111. 1111 1111. 1110 0000. 0000 0000  
0000 1010. 0000 1000. 1110 0000. 0000 0000

4. Переведем результат из двоичной системы счисления в десятичную систему счисления

0000 1010. 0000 1000. 1110 0000. 0000 0000 -> **10.8.224.0**

Поразрядная конъюнкция для единичных битов не меняет значение составляющей IP-адреса, а для нулевых битов дает значение 0. Следовательно, решение задачи можно упростить и  $255_{10} = 1111\ 1111_2$  не нужно переводить. Нулевая составляющая IP-адреса дает 0. И перевод в двоичный код имеет смысл только для третьего байта IP-адреса.

## IX. Комбинаторика

### Задание

5-буквенные слова из букв { С, У, Р, О, К }. Каждую букву нужно использовать ровно 1 раз, при этом нельзя ставить подряд две гласные или две согласные.

Сколько различных кодов может составить?

**Ответ: 12**

### Решение:

В предложенном алфавите согласных букв больше чем гласных, при этом они должны чередоваться между собой. Значит, слово не может начинаться с гласной. Получаем, что набор букв должен выглядеть как «С Г С Г С», где С - согласная, а Г - гласная.

Так как каждая буква должна появляться в слове только 1 раз, получаем число возможных вариантов:  $3 * 2 * 2 * 1 * 1 = 12$

## X. Адресация в электронных таблицах

### Задание

Выберите верную формулу, которая позволила рассчитать дополнительный балл (столбец “Доп. балл “) и записать результаты в столбец Е, для строк 2, 3, 4, 5, 6.

|   | А               | В          | С      | Д           | Е        |
|---|-----------------|------------|--------|-------------|----------|
| 1 | ФИО             | Математика | Физика | Информатика | Доп.балл |
| 2 | Иванов Сергей   | 66         | 66     | 77          | 0        |
| 3 | Петрова Мария   | 78         | 72     | 80          | 0        |
| 4 | Жуков Иван      | 92         | 83     | 93          | 1        |
| 5 | Бызов Петр      | 83         | 75     | 90          | 0        |
| 6 | Торопова Светла | 72         | 68     | 75          | 0        |

- 1) =ЕСЛИ(И(\$B\$2>80;\$C\$2>80;\$D\$2>80);1;0)
- 2) =ЕСЛИ(И(В2>80;С2>80;D2>80);1;0)
- 3) =ЕСЛИ(ИЛИ(В2>80;С2>80;D2>80);1;0)
- 4) =ЕСЛИ(ЕСЛИ(В2>80;ЕСЛИ(С2>80;ЕСЛИ(D2>80;1;0 );1));0)

**Ответ: 2**

**Решение:**

Используем функцию =ЕСЛИ() с дополнительной функцией «И»  
 =ЕСЛИ ( И ( условие; другое условие); значение, если ИСТИНА; значение, если ЛОЖЬ)

Дополнительные баллы вычисляются в столбце Е. измене

Для получения баллов в столбце Е нужно сравнить значения в столбцах В, С, D на больше значения 80.

Проверяем три условия  $B2 > 80$ ,  $B2 > 80$ ,  $C2 > 80$  , используя функцию «И».

Если все три условия истины, то в столбец Е записываем «1», если хотя бы одно из условий ложно записываем «0».

=ЕСЛИ ( И (  $B2 > 80$ ;  $B2 > 80$ ;  $C2 > 80$ ); 1; 0 )

Далее формула из ячейки D2 копируется в ячейки с D3 по D6. Так значения в формуле меняются, мы должны использовать относительную адресацию, без (\$), не используем абсолютную адресацию – когда адрес ячейки, содержащий исходное значение, не меняется. Относительная адресация - это изменяющийся при копировании и перемещении формулы адрес ячейки, содержащий исходное данное.

## XI. Количество информации

**Задание**

Необходимо придумать пароль длиной ровно 11 символов. В пароле можно использовать десятичные цифры и 12 различных символов местного алфавита, причем все буквы используются в двух начертаниях – строчные и прописные. Каждый символ кодируется одинаковым и минимально возможным количеством бит, а каждый пароль – одинаковым и минимально возможным количеством байт. Определите объем памяти в байтах, необходимый для хранения 60 паролей.

Выберите один из вариантов ответа:

- 1) 720
- 2) 660
- 3) 600
- 4) 540
- 5) 360

**Ответ: 4**

**Решение:**

Количество символов используемых для кодирования пароля составляет:

10 цифр + 12 строчных букв + 12 прописных букв = 34 символа.

Количество информации несущий один символ равен 6 бит, берем ближайшую степень двойки  $2^6 = 64$ .

Количество информации заложенное в каждом символе равно 6, количество символов в пароле равно 11, следовательно для пароля  $11 * 6 = 66$  бит.

Каждый пароль равен 66 битам информации, но по условию задачи каждый пароль кодируется одинаковым и минимально возможным количеством байт, следовательно нам необходимо узнать сколько байт в 66 битах.

Если разделить 66 на 8 получится дробное число, а нам необходимо найти целое количество байт на каждый пароль, поэтому находим ближайший множитель числа 8, который превысит количество бит – это 9 байт ( $9 * 8 = 72$ ).

Каждый номер кодируется 9 байтами.

Паролей 60, для них потребуется  $60 * 9 = 540$  байт.

## ХII. Кодирование

**Задание**

Какое минимальное количество бит потребуется для кодирования русского алфавита и положительных чисел от 128 до 1024 включительно ?

**Ответ: 10**

**Решение:**

В русском алфавите 33 буквы.

От 128 до 1024 находится 897 положительных чисел.

Всего нужно закодировать  $897 + 33 = 930$  слов.

$$2^{10} = 1024 > 930$$

Минимальное количество бит равно 10.

### ХІІІ. Рекурсивные алгоритмы

#### Задание

Дан рекурсивный алгоритм:

```
void fun( int n ) {  
    cout << n;  
    if( n < 8 ) {  
        fun(n+2);  
        fun(n+3);  
    }  
}
```

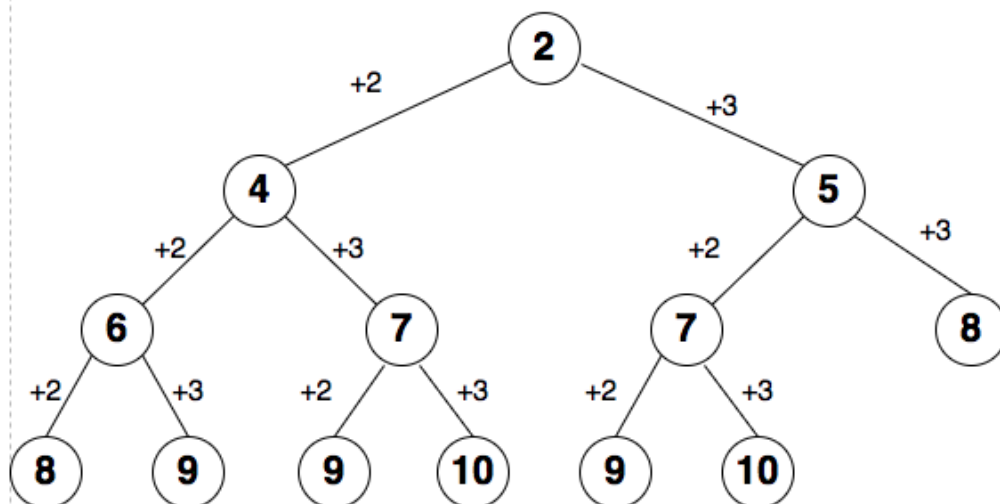
Найдите сумму чисел, которые будут выведены при вызове fun(2).

**Ответ: 94**

#### Решение:

Рекурсия - вызов функции из неё же самой, обращение функции самой к себе. Функция будет вызывать саму себя, но уже с другими параметрами, каждый раз увеличивая значение n на 2 и на 3 пока условие  $n < 8$  будет истинно.

Представим схему рекурсивных вызовов функции fun со значениями, при выполнении условия ( $n < 8$ ).



Сложим все числа. Результат равен 94.

### ХІV. Запросы для поисковых систем

#### Задание

В языке запросов к поисковому серверу для обозначения логической операции «или» используется символ «|». Для обозначения логической операции «и» используется символ «&». В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет. Считается, что все запросы выполнялись практически одновременно, так что

набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Какое количество страниц ( в тысячах) будет найдено по запросу:

***Кирпич & Блоки & Керамогранит ?***

| Запрос                          | Найдено страниц ( в тысячах) |
|---------------------------------|------------------------------|
| Кирпич & Керамогранит           | 1765                         |
| Блоки & Керамогранит            | 1211                         |
| (Кирпич   Блоки) & Керамогранит | 2534                         |

Какое количество страниц ( в тысячах) будет найдено по запросу:

***Кирпич & Блоки & Керамогранит ?***

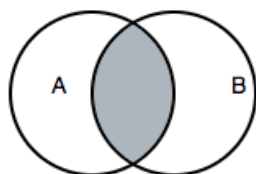
В ответе запишите целое число.

**Ответ: 442**

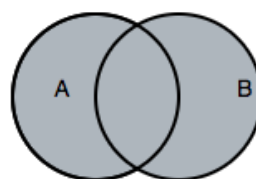
**Решение:**

Диаграмма Эйлера-Венна - наглядное средство для работы со множествами. На диаграммах изображаются все возможные варианты пересечения множеств.

Конъюнкция – логическое «И» - AND. Символ «&», «∧».



Дизъюнкция – логическое «ИЛИ» - OR. Символ «|», «∨».



Построим диаграмму Эйлера-Венна

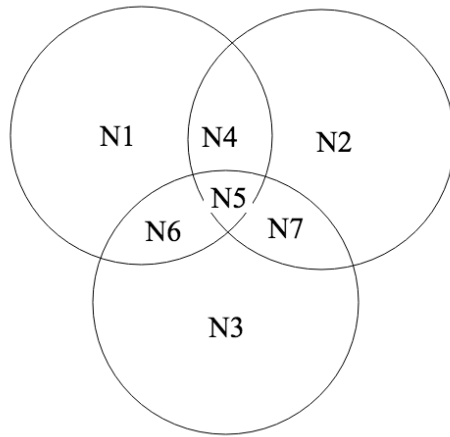
1. Определяем количество подмножеств - N1, N2, N3.
2. На диаграмме строим пересекающиеся множества - N1, N2, N3.
3. Выделяем области, количество которых равно количеству пересечений.

Обозначим:

N1 – Блоки

N2 – Керамогранит

N3 – Кирпич



Тогда:

$$N5 + N7 = 1765$$

$$N4 + N5 = 1211$$

$$N4 + N5 + N7 = 2534$$

$$N7 = 2534 - N4 - N5 = 2534 - 1211 = 1323$$

$$N5 = 1765 - N7 = 1765 - 1323 = 442$$

По запросу «Кирпич & Блоки & Керамогранит» найдено 442 страницы

## XV. Программирование

### Задание 1

Чему равно значение  $s$ , после выполнения кода ?

| Python  | C++   | Алгоритмический язык  |
|---|---|---|
| <pre>m=2 s=0 while m&lt;5:     s=s + m     m=m+1 print(s)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main(){     int m=2;     int s = 0;     while(m &lt; 5)     {         s = s + m;         m=m+1;     }     cout &lt;&lt; s&lt;&lt; endl; }</pre> | <pre>Начало Целые m, s m=2 s=0 нц пока m&lt;5     s=s+m     увеличиваем m на 1 кц ВЫВОД s КОН</pre> |

**Ответ: 9**

**Решение:**

Инициализируем две переменные целого типа  $m=2$  и  $s=0$

Цикл с предусловием будет выполняться пока условие ( $m < 5$ ) будет истинно, т.е. от 2 до 4 включительно. В цикле к значению  $s$  прибавляется значение  $m$  и значение счетчика  $m$  увеличивается на 1.

$s=0$

$m=2$

1 - ая итерация       $m < 5$      $s = 0 + 2 = 2$        $m = 2 + 1 = 3$

2 - ая итерация       $m < 5$      $s = 2 + 3 = 5$        $m = 3 + 1 = 4$

3 - ая итерация       $m < 5$      $s = 5 + 4 = 9$        $m = 4 + 1 = 5$

$m = 5$     Условие ( $m < 5$ ) ложно, т.к.  $m = 5$ . Осуществляется выход из цикла с предусловием, со значением  $s = 9$  и со значением  $m = 5$ . На экран выводится число 9.

## Задание 2

Чему равно значение  $k$ , после выполнения кода ?

| Python   | C++   | Алгоритмический язык  |
|--|---|---|
| <pre>i=0 k=0 a=[1,2,3,4,5,6,7,8,8,10] while i&lt;10:     if a[i] % 2 == 0 :         k = k + 1     i = i + 1 print("k=", k)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int a[] = { 1,2,3,4,5,6,7,8,8,10};     int i = 0;     int k = 0;     while (i &lt; 10)     {         if (a[i] % 2 == 0)             k = k + 1;         i = i + 1;     }     cout &lt;&lt; "k=" &lt;&lt; k &lt;&lt; endl; }</pre> | <p>Начало<br/>цел <math>i, k</math><br/><math>i=0</math><br/><math>k=0</math><br/>Массив<br/><math>a=1,2,3,4,5,6,7,8,8,10</math><br/>Нц пока <math>i &lt; 10</math><br/>    если эл. массива<br/>    четный,<br/>    то<br/>        <math>k = k + 1</math><br/>    значение счетчика<br/>    увеличиваем на 1<br/>Кц<br/>вывод <math>k</math><br/>Конец</p> |

**Ответ: 6**

## Решение:

Инициализируем две переменные целого типа  $i=0$  и  $k=0$

Объявляем и инициализируем массив из 10 целых чисел [ 1,2,3,4,5,6,7,8,8,10 ]

В цикле с предусловием , до тех пор пока условие ( $i < 10$ ) будет истинным,

выполняем проверку значения элемента массива на четность, используя арифметическую операцию остаток от деления - %.

Если значение массива четное, то увеличиваем значение счетчика k на 1 и увеличиваем значение i на единицу.

Из данного массива целых чисел [ 1,2,3,4,5,6,7,8,8,10 ], числа 2,4,6,8, 8,10 являются четными, при делении на 2 остаток равен нулю ( $i\%2==0$ ).

Значение счетчика k при выходе из цикла равно 6.

Обратите внимание – индексация массива начинается с нуля.

### Задание 3

Какое значение выведется на экран после выполнения кода программы, если ввести значения a = 10 и b=20 ?

| Python   | C++   | Алгоритмический язык   |
|--|---|--|
| <pre>def fun(a, b):     if a&gt;b and (a+b)%2==0:         print(a - b)     elif a&lt; b &amp;&amp;(a+b)%10==0:         print(b - a)     else:         print(a + b)  #главная функция a=int(input()) b=int(input()) fun(a, b)</pre> | <pre>#include &lt;iostream&gt; using namespace std; void fun(int a, int b) {     if (a&gt;b &amp;&amp; (a+b)%2==0)         cout&lt;&lt;a-b;     else         if(a&lt; b &amp;&amp;(a+b)%10==0)             cout&lt;&lt;b-a;         else             cout&lt;&lt;a + b; }  int main() {     int a, b;     cin&gt;&gt;a&gt;&gt;b;     fun(a, b);     return 0; }</pre> | <p>Функция с аргументами, (целые перемен. a и b)<br/>         Если a &gt; b и сумма (a+b) четная, то вывод разности a-b<br/>         Иначе<br/>         Если a &lt; b и остаток при делении на 10 суммы(a+b) равен 0, то вывод разности b-a<br/>         Иначе<br/>         вывод суммы a и b<br/>         Конец функции<br/>         Главная функция<br/>         Объявление целых a, b<br/>         Ввод a<br/>         Ввод b<br/>         Обращение к функции с параметрами a и b<br/>         Конец главной функции</p> |

**Ответ: 10**

### Решение:

В главной функции происходит обращение к функции fun с фактическими параметрами a = 10 и b = 20, которые вводятся с клавиатуры, как переменные целого типа.

Объявлена функция fun с двумя параметрами целого типа – int a, int b.

Функция принимает два значения a = 10 и b = 20.

В функции условие, которое объединено логической операцией сложения «И - &&», (a > b && (a + b)%2==0). Если хотя бы одно из этих условий ложно, то все условие ложно и управление переходит ко второму оператору сравнения. Если оба условия истинны, то выполняется вывод на экран разности (a - b). Иначе происходит второе сравнение, в котором два условия объединены логической операцией сложения «И - &&» - (a < b и (a + b)%10==0). Сравниваются значения (a < b) и сравнение (a + b) при делении на 10 на ноль. Если эти два условия истинны, то на экран выводится разность (b - a) и программа заканчивает выполнение, если хотя бы одно из условий ложно, то условие ложно и управление переходит на оператор иначе – else и на экран выводится сумма (a + b) и программа заканчивает выполнение.

Введено с клавиатуры два значения a = 10 и b = 20.

Первое условие (a>b) – ложно, второе условие истинно, но т.к. они объединены логическим “И”, то всё условие – ложно, то проверяем второе if, в котором два условия объединены логическим “И” - (a < b &&(a + b)%10==0). Здесь оба условия истинны, следовательно выполняем действие вывод на экран (b - a) . Результат вывода – 10.

#### Задание 4

Чему равен результат после обращения к функции f(5) ?

| Алгоритмический язык   | Python   | C++   |
|--|--|---|
| Функция с параметром n<br>Если n=1, возврат n<br>Иначе возврат и<br>обращение к функции с<br>аргументом (n-1)*n<br><br>Главная функция<br>Вывод на экран значения при<br>обращении к функции с<br>аргументом 5 | <pre>def f(n):     if n==1:         return n     else:         return n*f(n-1)  #главная функция print(f(5))</pre> | <pre>#include &lt;iostream&gt; using namespace std; int f(int n) {     if (n==1) return n;     else return n*f(n-1); } int main() {     cout&lt;&lt;f(5);     return 0; }</pre> |

**Ответ:120**

#### Решение :

Объявлена функция f с формальным параметром n – def f(n). Функция должна вернуть в точку вызова значение целого типа – return n или return n\*f(n-1). В функции если n не равно 1 происходит обращение функции к самой себе, но уже с параметром - f(n-1) – это рекурсия.

В главной функции происходит обращение к функции  $f$  с фактическим параметром  $5 - f(5)$ . Функция принимает значение  $n = 5$  и проверяет это значение. При рекурсии обязательно должно быть условие выхода из функции, иначе может произойти заикливание.. В нашем примере условие выхода – это проверка значения  $n$  на 1. Если значение  $n = 1$ , то происходит возврат в главную функцию со значение равным  $n - \text{return } n$ . Это обязательное условие выхода из рекурсивной функции. Функция обращается сама к себе с аргументом меньшим на единицу, но при этом умножая предыдущее значение на последующее, до тех пор пока  $n$  не станет равным 1 – это РЕКУРСИЯ.

Рекурсия - вызов функции из неё же самой, обращение функции самой к себе. Функция будет вызывать саму себя , но уже с другими параметрами, до условия выхода.

$$5 * f(4 * f(3 * f(2 * f(2 - 1)))) = 5 * 4 * 3 * 2 = 120$$