

**Открытая олимпиада школьников (информатика)
(№66 Перечня олимпиад школьников, 2021/2022 уч.год)**

Содержание

Содержание	1
Задания для 11 класса	2
Заключительный этап.....	2
Отборочный этап. Первый тур	15
Отборочный этап. Второй тур.....	21
Задания для 9 и 10 класса	26
Заключительный этап, 10 класс	26
Заключительный этап, 9 класс	36
Отборочный этап. Первый тур	44
Отборочный этап. Второй тур.....	48
Задания для 5–8 класса	52
Заключительный этап.....	52
Отборочный этап. Первый тур	62
Отборочный этап. Второй тур.....	66

Задания для 11 класса

Заключительный этап (приведен один из вариантов заданий)

1. Кодирование информации. Системы счисления (3 балла)

[Двоичный ряд]

Петя строит ряд двоичных чисел следующего вида:

$0.(01)_2, 0.0(01)_2, 0.00(01)_2, \dots$

Каждый член ряда представляет собой бесконечную периодическую дробь. Для получения каждого следующего числа Петя берет предыдущее число и вставляет один ноль после разделителя целой и дробной части.

Петя взял первые 30 чисел этого ряда и сложил их, записав результат в шестнадцатеричной системе счисления. Посчитайте сумму цифр в дробной части получившегося числа. В ответе укажите целое число в десятичной системе счисления. Например, если запись суммы чисел в шестнадцатеричной системе счисления будет равна $1,ABC_{16}$, то в ответ нужно записать 33.

Ответ: 78

Решение:

Рассмотрим первый член ряда и запишем равенство:

$$A: 0.(01)_2 = X_{10}$$

Домножим обе части равенства на $100_2 = 4_{10}$

$$B: 1.(01)_2 = (4 \cdot X)_{10}$$

Вычтем из равенства B равенство A.

$$B-A: 1.(01)_2 - 0.(01)_2 = (4 \cdot X)_{10} - X_{10}$$

$$1_2 = 1_{10} = (3 \cdot X)_{10}$$

$$X = 1/3$$

Обратим внимание, что вставка одного нуля после разделителя целой и дробной части эквивалентна делению числа на основание системы счисления. То есть, каждый следующий член ряда получается умножением предыдущего на $1/2$. Следовательно, мы имеем геометрическую прогрессию:

$1/3, 1/6, 1/12, \dots$

Сумма её первых 30 членов может быть получена с использованием известной формулы:

$$(1 - q^{n+1}) / (1 - q)$$

Подставив $q = 1/2$ и проведя несложные алгебраические преобразования, получим:

$$(2^n - 1) / (3 \cdot 2^{n-1})$$

Теперь нам нужно вычислить значение при $n=30$ и представить в двоичной системе счисления, из которой можно будет уже перевести в шестнадцатеричную. Это можно сделать различными способами. Один из вариантов, проанализировать значения числителя и знаменателя для нескольких значений n . Например, с помощью электронной таблицы:

n	числитель	знаменатель
1	1	3
2	3	6
3	7	12
4	15	24
5	31	48
6	63	96
7	127	192
8	255	384
9	511	768
10	1023	1536

Обратим внимание, что при четных значениях n (а 30 – четное число), числитель всегда кратен 3 (это можно доказать, например, через формулу для $a^{2n} - b^{2n}$), а значит для таких значений можно поделить числитель и знаменатель на 3. Построим новые значения числителя и знаменателя, поделенные на 3 для первых нескольких четных значений n и запишем их в двоичной системе счисления:

n	числитель	знаменатель	числитель в дв. с.с.	знаменатель дв. с.с.
2	1	2	1	10
4	5	8	101	1000
6	21	32	10101	100000
8	85	128	1010101	10000000
10	341	512	101010101	1000000000

Заметим, что тогда значения вычисляемой нами суммы для этих значений n в двоичной и шестнадцатеричной системах счисления будут следующими:

п	числитель в дв. с.с.	знаменатель дв. с.с.	сумма членов в дв. с.с.	дробная часть суммы членов в шестнадц. с.с.
2	1	10	0.1	8
4	101	1000	0.101	A
6	10101	100000	0.10101	A8
8	1010101	10000000	0.1010101	AA
10	101010101	1000000000	0.101010101	AA8

Как видно из таблицы, запись искомой суммы членов ряда в шестнадцатеричной системе счисления для значений n кратных 4 будет состоять из $n/4$ цифр A после разделителя целой и дробной части, а для n не кратных 4 – из $(n-2)/4$ цифр A и одной цифры 8 после разделителя целой и дробной части. То есть, для $n=30$ это будет 0,AAAAAAAAA8 и сумма цифр будет $(30-2)/4 \cdot 10 + 8 = 78$.

Заметим, что можно вычислить и непосредственно значения числителя и знаменателя для дроби $(2^{30}-1)/(3 \cdot 2^{30-1})$, поделить их на 3 и перевести в шестнадцатеричную систему, но это потребует достаточно трудоемких вычислений.

Альтернативное решение после того, как определили, что необходимо посчитать сумму членов геометрической прогрессии:

$$1/3 + 1/6 + 1/12 + \dots + 1/(3 \cdot 2^{29}) = (1/3) \cdot (1 + 1/2 + 1/(2^2) + 1/(2^3) + \dots + 1/(2^{29}))$$

Сгруппируем по парам

$$= (1/3) \cdot ((1 + 1/2) + (1/(2^2) + 1/(2^3)) + \dots + (1/(2^{28}) + 1/(2^{29}))) =$$

$$= (1/3) \cdot (3/2 + 3/(2^3) + \dots + 3/(2^{29}))$$

выносим 3 и сокращаем

$$= 1/2 + 1/(2^3) + 1/(2^5) + \dots + 1/(2^{29})$$

Получается сумма отрицательных нечетных степеней 2.

В двоичной системе это равно 0.10101010...1

Всего 29 знаков в дробной части. Переводим в 16-ую систему – по 4 знака от запятой ($29 = 4 \cdot 7 + 1$)

$$= 0.AAAAAAAAA8$$

Следовательно сумма цифр равна $10 \cdot 7 + 8 = 78$

2. Кодирование информации. Объем информации (2 балла)

[RLE]

Петя сохраняет в память текст. Известно, что алфавит текста составляет 32 символа. Петя использует равномерное кодирование и сохраняет в память коды символов, используя минимально возможное одинаковое для кодов всех символов количество бит.

Вася и Таня изучают кодирование длин серий run-length encoding (RLE) и пытаются его применить к тексту Пети для того, чтобы он занимал меньше места в памяти.

В случае RLE кодирования весь текст представляется как расположенные друг за другом непересекающиеся последовательности из одинаковых символов так, что символы в двух соседних последовательностях отличаются. При этом в памяти сохраняется для каждой последовательности два значения: длина последовательности и код повторяющегося в ней символа. Для записи кода символа используется минимальное возможное одинаковое для кодов всех символов количество бит. Для записи длины последовательности используется минимально возможное одинаковое для всех возможных значений этого параметра количество бит.

Вася обнаружил, что максимальная длина последовательности из одинаковых символов, которая встречается в тексте, равна M и решил, что в строке могут встречаться все возможные длины последовательностей, не превышающие M . Исходя из этого предположения, он определил количество бит, необходимое для записи длины последовательности и, сохранив в память текст Пети с помощью кодирования RLE, обнаружил, что он занимает ровно 208 байт памяти.

Также Вася обнаружил, что максимальная длина последовательности из одинаковых символов ровно в X раз меньше общего количества символов в тексте Пети.

Таня проанализировала текст более внимательно и обнаружила, что количество различных длин последовательностей, которые встречаются в тексте ровно в 16 раз меньше, чем длина максимальной последовательности. Исходя из этого, она определила другое количество бит, необходимое для записи длины последовательности и, сохранив в память текст Пети с помощью кодирования RLE, обнаружила, что он занимает ровно 144 байта памяти.

Определите значение X , если известно, что текст, сохраненный Петей, занимает на 12272 байта больше памяти, чем текст, сохраненный Васей. Если таких значений несколько, определите минимальное из них. В ответе укажите целое число.

Ответ: 78

Решение:

Поскольку текст, сохраненный Васей, занимает 208 байт и это на 12272 байта меньше, чем текст, сохраненный Петей, объем памяти, который занимает текст, сохраненный Петей, составляет $12272 + 208 = 12480$ байт.

Для сохранения кода одного символа при равномерном кодировании для алфавита из 32 символов потребуется $\log_2(32) = 5$ бит.

Следовательно, количество символов в тексте равно $12480 \cdot 8 / 5 = 19968$. И это значение представимо как произведение искомой нами величины X на пока неизвестное значение M . То есть $X = 19968 / M$.

Найдем значение M . Для этого составим уравнения исходя из известным нам значений объемов памяти, занимаемых при кодировании RLE Васей и Таней.

Обратим внимание, что поскольку по условию требуется найти минимальное значение X , нам требуется найти максимальное значение M .

Пусть N – количество последовательностей, на которое разбивается текст Пети, тогда:

$$\text{Вася: } N \cdot (\log_2(M) + \log_2(32)) = 208 \cdot 8$$

$$\text{Таня: } N \cdot (\log_2(M/16) + \log_2(32)) = 144 \cdot 8$$

Поскольку мы ищем максимальное значение M , мы можем считать, что в уравнениях стоят именно значения $\log_2(M)$, а не их округления в большую сторону до ближайшего целого числа.

Вычтем из первого уравнения второе:

$$N \cdot (\log_2(M) + \log_2(32)) - N \cdot (\log_2(M/16) + \log_2(32)) = (208 - 144) \cdot 8$$

$$N \cdot ((\log_2(M) + \log_2(32)) - (\log_2(M/16) + \log_2(32))) = 512$$

$$N \cdot (\log_2(M) + \log_2(32) - \log_2(M/16) - \log_2(32)) = 512$$

$$N \cdot (\log_2(M) - \log_2(M/16)) = 512$$

$$N \cdot (\log_2(M / (M/16))) = 512$$

$$N \cdot (\log_2(16)) = 512$$

$$N \cdot 4 = 512$$

$$N = 128$$

Подставим найденное значение N в первое уравнение:

$$128 \cdot (\log_2(M) + \log_2(32)) = 208 \cdot 8$$

$$128 \cdot (\log_2(M) + 5) = 1664$$

$$\log_2(M) + 5 = 1664 / 128$$

$$\log_2(M) = 8$$

$$M = 256$$

$$\text{Таким образом, } X = 19968 / 256 = 78$$

3. Основы логики (2 балла)

[Много чисел]

Логический преобразователь работает следующим образом:

1. На вход преобразователю подаётся целое неотрицательное число N , меньшее 2^{64}_{10} , записанное в шестнадцатеричной системе счисления. Запись числа переводится в двоичную систему счисления и разбивается на 16 групп по 4 разряда (при необходимости в начале записи дописываются незначащие нули).
2. Двигаясь справа налево по очереди берется каждая группа двоичных разрядов и представляется как последовательность из четырех логических переменных $X_0X_1X_2X_3$, так, что единичное значение двоичного разряда соответствует значению "истина", а нулевое - значению "ложь". Например, в числе 0...010111 первая такая последовательность будет $X_0 = \text{«ложь»}$, $X_1 = \text{«истина»}$, $X_2 = \text{«истина»}$, $X_3 = \text{«истина»}$.
3. Полученные последовательности логических переменных по очереди подставляются в логическое выражение:
$$(X_0 \vee X_1 \wedge X_2) \wedge (X_1 \vee X_2 \wedge X_3) \wedge (X_2 \vee X_3 \wedge X_0)$$

и вычисляется его значение.

4. Вычисленные значения преобразуются в двоичные разряды (также «истина» = 1, а «ложь» = 0) и образуют двоичную последовательность, заполняемую от младшего к старшему разряду.
5. Результатом работы преобразователя является шестнадцатиразрядное двоичное число (запись может содержать незначащие нули).

На вход преобразователю подали запись числа, равного FEDCBA9876543210₁₆ и получили некоторое число M . Сколько всего существует таких чисел, которые можно подать на вход преобразователю и получить в результате такое же число M ? Поскольку число может получиться достаточно большое, посчитайте и укажите в ответе сумму его цифр при записи в десятичной системе счисления.

Ответ: 27

Решение:

Построим таблицу истинности для заданного в условии выражения. Это можно сделать как вручную, так и написав программу. Например, так:

```
N=int('FEDCBA9876543210',16)
```

```
tab=[]
```

```
for i in range(16):
```

```
    num=N%16
```

```
    st=[]
```

```
    for j in range(4):
```

```
        st.append(num%2)
```

```
        num//=2
```

```
    F=1 if (st[3]==1 or st[2]==1 and st[1]==1) and (st[2]==1 or st[1]==1 and st[0]==1) and (st[1]==1 or st[0]==1 and st[3]==1)
```

```
else 0
```

```
    tab.append([st[::-1],F])
```

```
    N//=16
```

```
for i in tab:
```

```
    print(i)
```

X ₀	X ₁	X ₂	X ₃	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Заметим, что в таблице 10 строк имеют ложное значение и 6 строк имеют истинное значение.

Обратим внимание, что преобразователь, разбивая двоичную запись на 16 групп по 4 разряда получает двоичные записи отдельных цифр шестнадцатеричного числа (при необходимости, дополненного незначащими нулями). Эти записи можно интерпретировать как значения логических переменных в строках таблицы истинности. Приведенное в условии число FEDCBA9876543210₁₆ содержит все возможные шестнадцатеричные цифры, расположенные по убыванию, следовательно, если перевести это число в двоичную систему счисления и начать рассматривать группы по 4 разряда с конца к началу записи, как указано в условии, мы получим все комбинации значений логических переменных, то есть все строки построенной выше таблицы истинности. Тогда число М получается из последнего столбца как 1110100011000000₂ = 59584₁₀.

Значение выражения для каждой строки таблицы истинности получается независимо и может быть истинно или ложно. Тогда, получается, что каждая цифра шестнадцатеричного числа даёт либо 0 либо 1 в очередном по порядку следования двоичном разряде числа М. А из таблицы истинности мы знаем какая цифра какое значение даёт и в каком разряде должно быть какое значение. Следовательно, любая цифра, которая даст значение 1 может быть на любой позиции, на которой должно получиться значение 1, а любая цифра, которая даст значение 0 может быть на любой позиции, на которой должно получиться значение 0. Теперь вспомним наблюдение, которое мы записали после таблицы истинности и выведем выражение для вычисления количества подходящих чисел: $10^{10} \cdot 6^6 = 46656000000000$. Сумма цифр в этом числе равна 27, что и является правильным ответом.

4. Алгоритмизация и программирование. Формальный исполнитель (1 балл)

[Лабиринт]

Дан лабиринт, размером 10 на 10 клеток:

```

* . * * * * * * *
* . . . . . . *
* . * . * * . *
* . * . . . * . *
* . * . * * . *
* . * . * * . *
* . * . . . * . *
* . * * * * * . *
* . . . . . . *
* * * * * * *

```

Каждая клетка содержит один из двух символов: «*» означает непреодолимое препятствие, а «.» свободная клетка. В одну из свободных клеток помещают шарик. Если ниже этой клетки есть еще свободные клетки, шарик падает, пока не наталкивается на непреодолимое препятствие. Далее лабиринт поворачивают следующим образом:

1. На 90 градусов по часовой стрелке
2. На 90 градусов по часовой стрелке
3. На 90 градусов по часовой стрелке
4. На 90 градусов против часовой стрелки.

После каждого поворота, если ниже той клетки, в которой находится шарик, есть еще свободные клетки, шарик падает, пока не наталкивается на непреодолимое препятствие. После этого можно осуществлять следующий поворот.

Сколько существует клеток, в которые можно исходно поместить шарик так, что в результате выполнения указанных поворотов (возможно до их завершения), он выкатится из лабиринта? В ответе укажите целое число.

Ответ: 43

Решение:

Можно перебрать точки вручную, но в этом случае легко допустить ошибку по невнимательности. Можно реализовать алгоритм в виде программы. Ниже приведен пример на языке Python. Программа избыточна, поскольку демонстрирует также траектории для всех найденных точек, что может быть удобно для анализа решения задания. Для нахождения только ответа на задание её можно упростить.

```

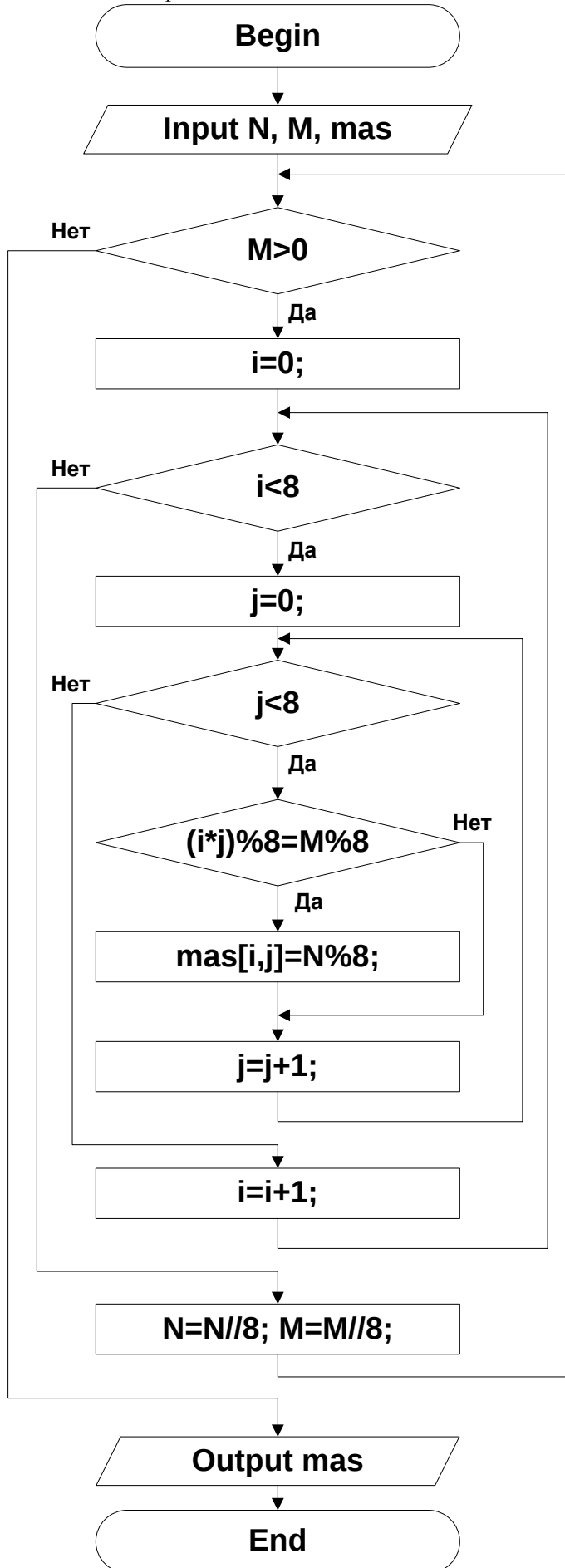
Path='RRRL'
ans=0
for x in range(10):
    for y in range(10):
        A=[]
        A.append(list("*.*****"))
        A.append(list("*.*****"))
        A.append(list("*.***.*.*"))
        A.append(list("*.***.*.*"))
        A.append(list("*.***.*.*"))
        A.append(list("*.***.*.*"))
        A.append(list("*.***.*.*"))
        A.append(list("*.***.*.*"))
        A.append(list("*.***.*.*"))
        A.append(list("*****"))
        if A[x][y]!='*':
            C=[x,y]
            A[C[0]][C[1]]='O'
            while A[C[0]+1][C[1]]!='*':
                C[0]+=1
                A[C[0]][C[1]]='o'
            D=0 #Направление: 0 - вниз, 1 - вправо, 2 - вверх, 3 - влево
            out=False
            for i in range(len(Path)):
                if not out:
                    D=(D+1)%4 if Path[i]=='R' else (D+3)%4
                    if D==0:
                        while A[C[0]+1][C[1]]!='*':
                            C[0]+=1
                            A[C[0]][C[1]]='o'
                    elif D==1:
                        while A[C[0]][C[1]+1]!='*':
                            C[1]+=1
                            A[C[0]][C[1]]='o'
                    elif D==2:
                        while A[C[0]-1][C[1]]!='*':
                            C[0]-=1
                            A[C[0]][C[1]]='o'
                        if C==[0,1]:
                            print('Finish from', x, y)
                            for i in A:
                                print("".join(i))
                            out=True
                            ans+=1
                            break
                else:
                    while A[C[0]][C[1]-1]!='*':
                        C[1]-=1
                        A[C[0]][C[1]]='o'
print(ans)

```

5. Алгоритмизация и программирование. Анализ алгоритма, заданного в виде блок-схемы (2 балла)

[Странная симметрия]

Дана блок-схема алгоритма.



На вход алгоритму подаются два целых положительных числа N и M и двумерный массив mas размером 8×8 элементов инициализированный нулями.

Известно, что на вход было подано число $M=8143989$. Определите минимальное значение числа N , такое, что на выходе будет массив mas со следующими значениями:

5	5	5	5	5	5	5	5
5	2	3	7	4	0	1	6
5	3	4	1	5	3	4	1
5	7	1	2	4	6	3	0
5	4	5	4	5	4	5	4
5	0	3	6	4	2	1	7
5	1	4	3	5	1	4	3
5	6	1	0	4	7	3	2

В ответе укажите целое число. Если такого числа не существует, запишите в ответ NULL.

Примечания:

1. При обращении к элементам массива первый индекс означает номер строки, а второй – номер столбца, нумерация строк и столбцов начинается с нуля.
2. Операция $A \% B$ означает вычисление остатка при целочисленном делении A на B , а операция $A // B$ означает вычисление частного при целочисленном делении A на B .

Ответ: 16434824

Решение:

Проанализируем алгоритм. Во внешнем цикле число M разбирается на цифры его записи в восьмеричной системе счисления от младшего разряда к старшему. Обратим внимание, что с числом N поступают аналогично. Тогда на каждом шаге внешнего цикла значения $M \% 8$ и $N \% 8$ – это очередные цифры восьмеричных записей чисел M и N , соответственно. Для каждой пары цифр чисел M и N двумя вложенными циклами перебираются все элементы двумерного массива. При этом, если для некоторого элемента выполняется условие, что остаток от деления произведения индексов $(i*j)$ этого элемента равен текущей цифре восьмеричной записи числа M , в этот элемент записывается текущая цифра восьмеричной записи числа N . Цикл продолжается, пока не будут перебраны все цифры в числе M .

Переведем число M в восьмеричную систему счисления и получим, что $8143989_{10} = 37042165_8$. Обратим внимание, что в записи числа содержатся все возможные восьмеричные цифры и ровно по одному разу каждая.

Теперь построим таблицу значений остатков $(i*j) \% 8$. Это можно сделать вручную, но удобнее воспользоваться программированием или электронными таблицами. Например, используя формулу $=\text{ОСТАТ}(\$A2*B\$1;8)$ в ячейках диапазона $B2:I9$:

	A	B	C	D	E	F	G	H	I	J
1		0	1	2	3	4	5	6	7	
2	0	0	0	0	0	0	0	0	0	
3	1	0	1	2	3	4	5	6	7	
4	2	0	2	4	6	0	2	4	6	
5	3	0	3	6	1	4	7	2	5	
6	4	0	4	0	4	0	4	0	4	
7	5	0	5	2	7	4	1	6	3	
8	6	0	6	4	2	0	6	4	2	
9	7	0	7	6	5	4	3	2	1	
10										

Обратим внимание, что там, где в построенной таблице встречается значение 5, в двумерном массиве из условия стоит младшая цифра восьмеричной записи числа N , то есть она равна 0. Аналогично определим остальные цифры. Обратим внимание, что для этого достаточно проанализировать вторую строку или второй столбец. В результате получим восьмеричное число 76543210_8 . Переведем его в десятичную систему счисления и получим ответ 16434824 .

6. Телекоммуникационные технологии (2 балла).

[Wildcard Mask]

В IPv4 сетевой адрес представляет собой 4-байтное число, записанное в десятичной форме, где отдельные байты разделены точкой. В IP-адресе выделяются две части – адрес сети и адрес узла. Деление происходит с помощью маски – 4-х байтного числа, которое поставлено в соответствие IP-адресу. Маска содержит двоичные 1 в тех разрядах IP-адреса, которые определяют адрес сети и двоичные 0 в тех разрядах IP-адреса, которые определяют адрес узла.

Кроме обычной маски существует еще wildcard mask. Эти маски часто применяются для написания правил фильтрации в межсетевых экранах (firewall) для того, чтобы определить применяется ли правило к IP-адресу из проверяемого пакета или нет. Wildcard mask – 4-байтное число, которое содержит двоичные единицы в тех разрядах IP-адреса, которые могут в IP-адресе содержать разные значения, и нули в тех разрядах адреса, которые должны совпадать с заданными в правиле фильтрации. Wildcard mask не обязана содержать подряд идущие единицы или нули.

Межсетевой экран — программный или программно-аппаратный элемент компьютерной сети, осуществляющий контроль и фильтрацию проходящего через него сетевого трафика в соответствии с заданными правилами. В этой задаче мы

будем использовать синтаксис команд для оборудования cisco. Реализована отдельная процедура составления правил. Проверка правил идет сверху вниз по списку до первого срабатывания.

Рассмотрим пример, в котором используются обычные mask и wildcard mask. Будем считать, что для определения типа маски используется её старший бит: 1 – обычная маска, а 0 – wildcard mask. В примере опущены команды, необходимые для переключения контекстов управления и активации списка правил.

правило 1. создать список контроля с именем 123 и добавить в него правило, пропускающее пакеты с одного адреса 192.168.0.11 на любые адреса. То, что правило работает для одного адреса, видно из использованной маски, состоящей целиком из 32-х единиц.

```
access-list 123 permit ip 192.168.0.11 255.255.255.255 any
```

правило 2. добавить в список 123 правило, запрещающее передачу пакетов со всех нечетных адресов сети 192.168.0.0 255.255.255.0 по любым направлениям

```
access-list 123 deny ip 192.168.0.1 0.0.0.254 any
```

Рассмотрим подробнее правило с wildcard mask. Предположим, на порт поступает пакет с адреса 192.168.0.127. И надо проверить, применять ли правило 2 к этому адресу. Представим в двоичном формате проверяемый адрес, адрес-образец из правила и wildcard mask.

192.168.0.127 – проверяемый адрес	11000000.10101000.00000000.01111111
192.168.0.1 – адрес образец	11000000.10101000.00000000.00000001
0.0.0.254 – wildcard mask	00000000.00000000.00000000.11111110

Желтым отмечены разряды, которые должны совпадать в адресе-образце из правила и проверяемом адресе. Номера этих разрядов определяются по wildcard mask.

Зеленым цветом выделены разряды, которые могут отличаться у проверяемого адреса и адреса-образца из правила (в этих разрядах wildcard mask содержит 1).

Очевидно, что для адреса 192.168.0.127 правило применится, а, например, для адреса 192.168.0.128 это правило не применится (не совпадет последний бит).

Также отметим, что для адреса 192.168.0.11 это правило не будет применено, потому что для этого адреса будет использовано правило 1, находящееся раньше по списку.

Задание

Ниже приведен список команд, составляющий исходный набор правил. Требуется сократить количество правил, а, следовательно, и время проверки, с использованием **только** wildcard mask. Есть набор возможных вариантов команд, которые могут быть включены в новый набор правил. Выберите из этого набора нужные команды и составьте из них набор правил минимально возможной длины, такой, что он будет полностью соответствовать исходному набору правил. Приведите в ответе через пробел номера из набора возможных вариантов команд. Номера указывайте в нужном порядке, там, где это необходимо. Если подходящих вариантов порядка несколько, приведите тот, в котором номера идут по возрастанию. Если есть равнозначный выбор между несколькими правилами с одинаковой wildcard mask, выбирается правило с минимальным адресом-образцом.

Межсетевой экран непосредственно подключен к сети с адресом 172.21.25.144 и маской 255.255.255.240.

Список правил составляется **только** для реальных адресов указанной сети.

Исходный набор правил:

```
access-list 123 permit ip 172.21.25.145 255.255.255.255 any
access-list 123 permit ip 172.21.25.146 255.255.255.255 any
access-list 123 permit ip 172.21.25.147 255.255.255.255 any
access-list 123 permit ip 172.21.25.148 255.255.255.255 any
access-list 123 permit ip 172.21.25.149 255.255.255.255 any
access-list 123 permit ip 172.21.25.150 255.255.255.255 any
access-list 123 permit ip 172.21.25.151 255.255.255.255 any
access-list 123 deny ip 172.21.25.152 255.255.255.255 any
access-list 123 deny ip 172.21.25.153 255.255.255.255 any
access-list 123 deny ip 172.21.25.156 255.255.255.255 any
access-list 123 deny ip 172.21.25.157 255.255.255.255 any
access-list 123 permit ip 172.21.25.154 255.255.255.255 any
access-list 123 permit ip 172.21.25.155 255.255.255.255 any
access-list 123 permit ip 172.21.25.158 255.255.255.255 any
```

Возможные варианты команд:

1. access-list 123 permit ip 172.21.25.155 0.15.0.1 any
2. access-list 123 deny ip 172.21.25.152 0.0.0.110 any
3. access-list 123 permit ip 172.21.25.145 0.0.0.34 any
4. access-list 123 permit ip 172.21.25.154 0.0.1.250 any
5. access-list 123 permit ip 172.21.25.170 0.0.0.8 any
6. access-list 123 permit ip 172.21.25.144 0.0.0.7 any
7. access-list 123 permit ip 172.21.25.0 0.0.7.27 any
8. access-list 123 permit ip 172.21.25.0 255.255.255.240 any
9. access-list 123 deny ip 172.21.25.165 0.0.0.13 any
10. access-list 123 deny ip 172.21.25.152 0.0.0.5 any
11. access-list 123 deny ip 172.21.25.154 0.0.0.7 any
12. access-list 123 permit ip 172.21.25.154 0.0.0.5 any

Ответ: 6 10 12

Решение:

В условии указано, что межсетевой экран непосредственно подключен к сети с адресом 172.21.25.144 и маской 255.255.255.240 и список правил составляется **только** для реальных адресов указанной сети. Переведем этот адрес и маску в двоичную систему счисления:

Адрес сети: 10101100.00010101.00011001.10010000

Маска: 11111111.11111111.11111111.11110000

Поскольку маска содержит только 4 нуля, в указанной сети может быть только $2^4 - 2 = 14$ реальных узлов (помним, что еще два адреса зарезервированы для служебного адреса сети и адреса ограниченного широковещания и не могут быть назначены реальным узлам). Тогда все адреса реальных узлов указанной сети будут иметь первые три октета 172.21.25, а в последнем октете входить в диапазон от 145 до 158, то есть в двоичном виде отличаться только значениями младших четырех бит.

Заметим, что в исходном наборе правил также ровно 14 правил и для каждого адреса из определенного выше диапазона индивидуально определено значение «разрешен» (permit) или «запрещен» (deny).

Возьмем исходный набор правил и отсортируем его так, чтобы сначала шли все разрешающие правила, а потом запрещающие и чтобы в каждой группе правила были отсортированы по возрастанию значения в последнем октете адреса. Также посчитаем значения младших 4-х бит этих адресов, поскольку, как было отмечено выше, только они могут различаться для указанной сети.

	3	2	1	0	Действие
145	0	0	0	1	permit
146	0	0	1	0	permit
147	0	0	1	1	permit
148	0	1	0	0	permit
149	0	1	0	1	permit
150	0	1	1	0	permit
151	0	1	1	1	permit
154	1	0	1	0	permit
155	1	0	1	1	permit
158	1	1	1	0	permit
152	1	0	0	0	deny
153	1	0	0	1	deny
156	1	1	0	0	deny
157	1	1	0	1	deny

Теперь построим для предлагаемых вариантов правил следующую таблицу: младшие 4 бита последнего октета адреса-образца, последние 4 бита wildcard mask и в последних 4-х столбцах отметим значения тех битов адреса-образца, которые фиксируются с помощью wildcard mask. Правило 8 уберем сразу, поскольку оно не содержит wildcard mask.

№ правила	Последний октет адреса-образца и его младшие 4 бита					Действие	Последний октет wildcard mask и его младшие 4 бита					Фиксированные значения младших 4-х бит			
		3	2	1	0			3	2	1	0				
1	155	1	0	1	1	permit	1	0	0	0	1	1	0	1	
2	152	1	0	0	0	deny	110	1	1	1	0				0
3	145	0	0	0	1	permit	34	0	0	1	0	0	0		1
4	154	1	0	1	0	permit	250	1	0	1	0		0		0
5	170	1	0	1	0	permit	8	1	0	0	0		0	1	0
6	144	0	0	0	0	permit	7	0	1	1	1	0			
7	0	0	0	0	0	permit	27	1	0	1	1		0		
9	165	0	1	0	1	deny	13	1	1	0	1			0	
10	152	1	0	0	0	deny	5	0	1	0	1	1		0	
11	154	1	0	1	0	deny	7	0	1	1	1	1			
12	154	1	0	1	0	permit	5	0	1	0	1	1		1	

Заметим, что в исходном наборе правил есть и разрешающие и запрещающие правила. Значит, минимально возможное количество правил в новом наборе равно двум. Для запрещающих правил из таблицы видно, что они охватывают все комбинации из 4-х бит, у которых 1-й бит (если считать слева направо от 1) равен 1, а 3-й бит равен 0. По второй таблице определим, что этому соответствует правило 10. Разрешающие правила можно разбить на две группы. Для первой группы 1-й бит равен 0, и в исходных правилах есть все возможные комбинации, кроме (0,0,0,0), но эта комбинация является служебным адресом данной сети и не может быть назначена в адресе реального узла. По второй таблице определим, что для этой группы подойдет правило номер 6. Осталось найти правило для второй группы разрешающих правил. Заметим, что в эту группу входят все комбинации, в которых 1-й и 3-й биты равны 1 (кроме (1,1,1,1), которая будет соответствовать адресу ограниченного

широковещания, и не может быть назначена реальному узлу). Такие комбинации соответствуют правилу 12 из второй таблицы. Поскольку все три правила не имеют пересечений по адресам, их можно указывать в любом порядке, а, значит, в соответствии с условием их нужно указать в порядке возрастания. Тогда ответ будет 6 10 12.

7. Технологии обработки информации в электронных таблицах, сортировки и фильтрации данных (1 балл)

[Остатки]

Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D	E	F	G	H	I	J
1		2	3	5	6	10	15	30		
2		=СЧЁТЕСЛИ(B3:B102;"0")								
3		=ОСТАТ(\$A3;B\$1)							=СЧЁТЕСЛИ(B3:H3;"0")	
4										

Ячейку B2 скопировали во все ячейки диапазона C2:I2. Ячейку B3 скопировали во все ячейки диапазона B3:H102. Ячейку I3 скопировали во все ячейки диапазона I4:I102. Диапазон A3:A102 заполнили некоторым набором целых положительных чисел.

Известны значения некоторых ячеек, получившиеся после этого:

	A	B	C	D	E	F	G	H	I	J
1		2	3	5	6	10	15	30		
2		50	33	20	17	10	7	4		
3										
4										
5										

Определите значение в ячейке I2. В ответе укажите целое число.

Ответ: 27

Решение:

Обратим внимание на числа в диапазоне B1:H1. Сначала идут три простых числа (2, 3, 5), затем три пары их возможных произведений (2*3=6, 2*5=10, 3*5=15) и, наконец, произведение всех трех чисел (2*3*5=30).

Диапазон B3:B102 заполнен формулой =ОСТАТ(\$A3;B\$1). То есть, в столбце B этого диапазона будут остатки от деления чисел из диапазона A3:A102 на 2, в столбце C – остатки от деления на 3, в столбце D – остатки от деления на 5 и т.д.

Тогда в ячейках диапазона B2:H2 будут числа, означающие, сколько чисел из диапазона A3:A102 делятся на 2, сколько на 3, сколько на 5, сколько на 6, сколько на 10, сколько на 15 и сколько на 30. Обратим внимание, что эти числа можно интерпретировать как мощности трех пересекающихся множеств: множество чисел из диапазона A3:A102, которые делятся нацело на 2, множество чисел этого же диапазона, которые делятся нацело на 3 и множество чисел из этого диапазона, которые делятся нацело на 5. Формулы из диапазона I3:I102 подсчитывают для каждого числа количество делителей из набора {2, 3, 5}. Значение 0 в этом столбце будет только у тех чисел, которые не делятся ни на 2, ни на 3, ни на 5. Тогда в ячейке I2, значение которой нам нужно найти, будет количество таких чисел из диапазона A3:A102, которые не делятся ни на 2, ни на 3, ни на 5.

Посчитаем мощность объединения множеств, чисел из диапазона A3:A102, которые делятся на 2, 3 или 5. Поскольку мы знаем про все пересечения отдельных множеств, воспользуемся формулой включений-исключений и получим M=50+33+20-17-10-7+4=73. Всего в диапазоне A3:A102 ровно 100 чисел. Следовательно, 100-73=27 из них не делятся ни на 2, ни на 3, ни на 5. Значит, именно это число в ячейке I2.

8. Технологии программирования (2 балла)

[Биржа]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

В данной задаче вам требуется реализовать подсчет прибыли при торговле акциями.

Торги проходят в течении n дней, всего на рынке представлены акции m компаний, цены акций компании фиксированы в течение одного дня. Сделки бывают двух типов:

Купить x акций компании *comp*

Продать все акции компании *comp*

За каждую сделку надо заплатить 1% комиссии. Например, если купить 10 акций по 300 рублей, то суммарно заплатить придется 3030 рублей. Если же продавать 10 акций стоимостью 300 рублей каждая, то за них можно получить 2970 рублей.

Прибылью с продажи будем считать разность полученных при продаже денег и суммарно потраченных денег при покупках. Например, если 10 акций были куплены по 300 рублей, а затем еще 5 акций были куплены по 400 рублей, то в случае продажи по стоимости 500 прибыль составит: $15 \cdot 500 - 0.99 \cdot (10 \cdot 300 + 5 \cdot 400) = 7425 - (3030 + 2020) = 2375$ рублей. При этом акции могут быть проданы в убыток (за меньшую стоимость, чем были куплены), тогда прибыль с продажи будем считать отрицательной.

Суммарной прибылью в момент времени будем считать сумму всех прибылей с продаж, которые произошли до этого момента. Обратите внимание, что деньги, потраченные к этому моменту на покупку акций, которые еще не были проданы, не учитываются. Также будем считать, что у нас неограниченный бюджет для покупки акций, но рассматривать будем только прибыль. Число акций для покупки/продажи для каждой компании на рынке также не ограничено.

Вам даны k событий покупки/продажи. Необходимо найти минимальную суммарную прибыль среди всех моментов времени.

Формат входных данных

В первой строке входных данных содержится одно целое число t — число тестовых наборов ($1 \leq t \leq 30$).

Затем следуют t тестовых наборов. Каждый тестовый набор описывается следующим образом:

В первой строке тестового набора содержатся три целых числа n , m и k — число дней, в которые проходят торги, число компаний на рынке и число событий, соответственно ($1 \leq n, m \leq 100$, $1 \leq k \leq 1000$).

В следующих m строках записаны названия компаний и n чисел — стоимости акций компании в рублях в каждый из дней торгов. Названия компаний состоят из не более чем 10 строчных букв латинского алфавита и попарно различны. Стоимости акций — целые числа в диапазоне от 1 до 10^5 включительно.

В следующих k строках заданы события покупки/продажи в хронологическом порядке. Событие покупки задается в формате **<день> buy <число акций> <название компании>**, а событие продажи задается в формате **<день> sell <название компании>**. При этом **<день>** — целое число от 1 до n , а **<число акций>** — целое число от 1 до 1000. Гарантируется, что все события следуют в порядке неубывания дней и корректны, а именно нет продаж некупленных акций и покупок акций, которых нет на рынке.

Формат выходных данных

Для каждого тестового набора выведите в отдельной строке минимальную прибыль среди всех моментов времени, с относительной или абсолютной погрешностью не более 10^{-4} .

Пример

Стандартный ввод	Стандартный вывод
4 3 1 3 comp 300 400 500 1 buy 10 comp 2 buy 5 comp 3 sell comp 3 2 4 gazp 100 111 300 yndx 1000 1100 1111 1 buy 10 gazp 2 buy 1 yndx 3 sell yndx 3 sell gazp 3 1 3 comp 300 400 200 1 buy 10 comp 2 buy 5 comp 3 sell comp 2 2 3 bdn 100 100 nik 1 100 1 buy 300 bdn 1 buy 10 nik 2 sell nik	0 -11.11 -2080.0 0

Замечание

В первом тестовом наборе изначально до продаж суммарная прибыль равна 0, после первой продаже суммарная прибыль становится 2375 (случай разобран в примере).

Во втором тестовом наборе промежуточные прибыли равны 0, −11.11 (акция продана дороже, но комиссия больше разницы) и 1948.89.

В третьем тестовом наборе промежуточные прибыли равны 0 и −2080.

В четвертом тестовом наборе промежуточные прибыли равны 0 и 979.9, деньги, потраченные на непроданные акции, не учитываются.

Решение:

Для решения этой задачи необходимо написать ровно то, что просят в условии. Наиболее удобно это можно сделать, если воспользоваться ассоциативным массивом, он же `std::map` в C++ или `dict` в Python.

Тогда, обрабатывая события, будем для каждой компании поддерживать сколько денег мы потратили на покупку акций и сколько у нас акций этой компании есть в текущий момент. Также надо не забыть домножить на 1.01, чтобы учесть комиссию.

При обработке события продажи будем умножать количество имеющихся акций на текущую цену и на 0.99, потому что комиссия будет идти уже в минус. После чего добавит к текущей прибыли, разность полученных денег потраченных денег, которые мы как раз храним в ассоциативном массиве.

Соответственно далее остается только найти минимум/максимум «промежуточной» прибыли.

С примерами авторских решений можно ознакомиться в папке `solutions` в архиве задачи.

9. Технологии программирования (4 балла)

[Трансформация массива]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

Вам дано t пар массивов a_i и b_i равной длины.

За одну операцию модификации можно:

Поменять местами любые два элемента массива a_i , но каждый элемент массива может участвовать не более чем в одном обмене.

Прибавить к любому элементу массива a_i единицу. Данную операцию можно применять неограниченное число раз к любому элементу массива.

Для каждой пары массивов найдите минимальное число операций, которые необходимо применить к массиву a_i , чтобы получить массив b_i , или определите, что это невозможно.

Формат входных данных

В первой строке дано число t — число пар массивов ($1 \leq t \leq 40$).

В следующих $3t$ строках содержатся описания пар массивов. Каждая пара описывается тремя строками.

В первой из них дано число n_i — количество элементов в каждом массиве i -й пары ($1 \leq n_i \leq 10$). Во второй строке заданы n_i чисел $a_{i,j}$ — элементы массива a_i ($1 \leq a_{i,j} \leq 1000$). В третьей строке заданы n_i чисел $b_{i,j}$ — элементы массива b_i ($1 \leq b_{i,j} \leq 1000$).

Гарантируется, что сумма n_i по всем тестовым наборам не превосходит 150.

Формат выходных данных

Для каждой пары массивов выведите одно число — минимальное число операций, которые необходимо применить к массиву a_i , чтобы получить массив $b_{i,j}$, или −1, если для данной пары это невозможно.

Пример

Стандартный ввод	Стандартный вывод
6	-1
1	-1
2	1
1	-1
4	7
1 2 3 4	-1
2 4 1 2	
1	
1	
2	
5	
1 2 3 4 5	
1 5 1 1 1	
5	
1 2 3 4 5	
5 1 5 5 5	
3	
1 2 3	
3 1 2	

Решение

Данную задачу можно решать двумя способами: динамическим программированием или жадным подходом.

Для начала заметим, что если элемент массива a_i не участвует в обмене, то мы однозначно знаем сколько нужно операций, чтобы получить соответствующий элемент b_i (или что это невозможно). Также заметим, что после конкретного обмена мы также знаем сколько нужно операций изменения элемента.

Тогда, если решать динамическим программированием, можно сказать $dp[mask]$ — минимальное число операций, чтобы все элементы массива a_i с индексами соответствующими битовой маске $mask$, стали соответствующими элементами массива b_i . Решение очень похоже на поиск паросочетания минимального веса, в целом, тут и правда спрятан граф.

Для жадного решения нужно было доказать/поверить в то, что обмены довольно дешевые и их всегда выгодно делать, а затем перебирая элементы массива a_i в порядке возрастания находить оптимальную пару для обмена.

С примерами авторских решений можно ознакомиться в папке `solutions` в архиве задачи.

Отборочный этап. Первый тур (приведен один из вариантов заданий)

1. Кодирование информации. Системы счисления (2 балла)

[Три степени]

Вариант 1

Значение арифметического выражения

$$0,008_{16}^X + 0,04_{16}^Y + 0,2_{16}^Z$$

вычислили для значений $X=D$, $Y=7$, $Z=2$ и записали в системе счисления с основанием 16. Посчитайте сумму цифр в получившейся записи. В ответе укажите целое число в десятичной системе счисления.

Ответ: 16

2. Кодирование информации. Системы счисления (1 балл)

[Большие основания]

Вариант 1

Петя узнал про длинную арифметику и заинтересовался системами счисления с большими основаниями. Он обнаружил, что любое целое положительное число, большее 2, можно записать в некоторой системе счисления с основанием X как 11_X . Пете интересно, а можно ли по заданному числу определить все основания позиционных систем счисления, в которых запись этого числа будет состоять ровно из двух одинаковых цифр. Для эксперимента Петя решил взять десятичное число 2^X , где $X=151$. Определите для этого числа количество оснований позиционных систем счисления, в которых запись этого числа будет состоять ровно из двух одинаковых цифр. В ответе укажите целое число в десятичной системе счисления.

Ответ: 76

3. Кодирование информации. Количество информации. Кодирование текста (1 балл)

[Пары символов]

Вариант 1

Текст состоит из 2400 символов. Известно, что в тексте встречается ровно 100 различных символов. При сохранении текста в памяти сохраняются коды символов так, что для записи каждого кода используется минимально возможное, одинаковое для всех кодов символов количество бит.

Петя обнаружил, что весь текст можно представить в виде последовательности непересекающихся пар символов, причем в тексте встретится ровно K различных пар символов. Петя решил сохранять в памяти код каждой пары символов, используя для записи каждого кода минимально возможное, одинаковое для всех кодов пар символов количество бит.

Петя обнаружил, что в результате объем памяти, требующейся для сохранения текста, уменьшился ровно на 1050 байт. При каком минимальном значении K это возможно. В ответе укажите целое число.

Ответ: 65

4. Кодирование информации. Количество информации (3 балла)

[Квадраты]

Петя разрабатывает генератор QR-кодов. Пока он научился генерировать все квадратные изображения, размером 9 на 9 пикселей, каждый из которых окрашен черным или белым цветом. Полученные изображения он сохраняет в памяти друг за другом как последовательности из 81 бита. Петя решил, что если среди сгенерированных изображений встретится изображение, в котором присутствует квадрат, размером 8 на 8, состоящий только из черных пикселей, то такое изображение будет считаться бракованным. Бракованные изображения он решил сохранять отдельно. Определите, сколько памяти ему потребуется для этого. Ответ округлите в большую сторону до целого числа МБайт и запишите в ответ получившееся число.

Примечание: 1 Мбайт = 2^{20} байт.

Ответ: 6

5. Основы логики. Анализ логических функций (2 балла)

[Четыре выражения]

Вариант 1

Задана логическая функция от 5 переменных:

$$F(A, B, C, D, E) = (((\$1) \rightarrow (\$2)) \rightarrow (\$3)) \rightarrow (\$4)$$

Известно, что вместо обозначений $\$1$, $\$2$, $\$3$ и $\$4$ подставили следующие логические выражения:

1. A and not B
2. B and not C
3. C and not D
4. D and not E

Были использованы по одному разу все четыре логических выражения, но неизвестно, какому обозначению соответствует какое логическое выражение.

Для этой логической функции построили таблицу истинности: для всех комбинаций значений переменных A, B, C, D, E , расположенных в лексикографическом порядке вычислили значение логической функции и записали в виде столбца, обозначая истинное значение за 1, а ложное за 0. Получившиеся в столбце двоичные значения представили как разряды двоичного числа так, что первой строке (значениям A, B, C, D, E равным нулю), соответствует младший разряд, а последней строке (значениям A, B, C, D, E равным единице) соответствует старший разряд. В результате получилось число 3140533179. Определите, в каком порядке были записаны четыре логических выражения при задании логической функции F . В ответе

укажите номера логических выражений в порядке их следования в функции слева направо. Если такой логической функции не существует, в ответе запишите NULL.

Ответ: 1243

6. Основы логики. Упрощение логического выражения (2 балла)

[Вальс]

Упростите логическое выражение или укажите его результат (при его однозначности). Результат упрощения может содержать только операции инверсии, конъюнкции и дизъюнкции.

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

Комментарий по вводу ответа: операнды вводятся большими **латинскими** буквами; логические операции обозначаются, соответственно как **not**, **and** и **or**.

Скобки используются только для изменения порядка выполнения операций. Если порядок выполнения операций очевиден из их приоритетов – дополнительное использование скобок считается ошибкой.

При однозначном ответе – истинный ответ обозначается как 1, а ложный как 0.

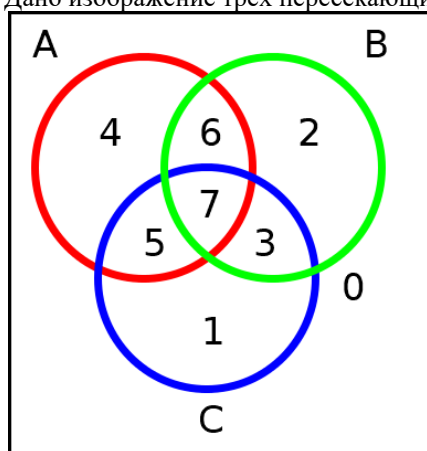
Пример записи ответа: (A or not B) and C

Ответ: not C or D or E || not C or E or D || D or not C or E || D or E or not C || E or not C or D || E or D or not C

7. Основы логики. Синтез выражения по логической схеме (1 балл)

[Три окружности]

Дано изображение трех пересекающихся окружностей:



На изображении выделены восемь непересекающихся пронумерованных областей (область, пронумерованная нулем, означает точки, не принадлежащие ни одной из окружностей).

Для этого изображения сформулированы три логических высказывания:

A = {точка находится внутри красной окружности}

B = {точка находится внутри зеленой окружности}

C = {точка находится внутри синей окружности}

Укажите в порядке возрастания через пробел все номера областей, для которых будет истинно следующее выражение:

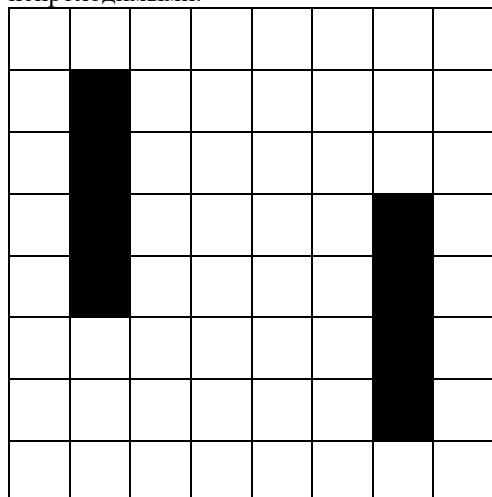
$(A \text{ xor not } B) \text{ xor } (B \text{ xor not } C)$

Ответ: 1 3 4 6

8. Алгоритмизация и программирование. Формальный исполнитель (3 балла)

[Загнанный в угол]

Дано клетчатое поле, размером 8 на 8 клеток. Часть из клеток, окрашенных на рисунке черным цветом, являются непроходимыми.



Робот-исследователь движется в соответствии со следующими правилами:

1. Изначально робот помещается в одну из клеток поля, не являющуюся непроходимой.
2. Робот не может перемещаться за пределы поля.
3. Робот не может перемещаться в непроходимую клетку.
4. Робот не может перемещаться в клетку, в которой он уже побывал.
5. Совершая очередной ход, робот пытается попасть в одну из соседних клеток, руководствуясь следующим алгоритмом:
 - a. Если робот может переместиться на одну клетку вниз, он перемещается на одну клетку вниз.
 - b. В противном случае, если робот может переместиться на одну клетку вправо, он перемещается на одну клетку вправо.
 - c. В противном случае, если робот может переместиться на одну клетку вверх, он перемещается на одну клетку вверх.
 - d. В противном случае, если робот может переместиться на одну клетку влево, он перемещается на одну клетку влево.
 - e. Если робот не может переместиться на одну клетку ни вверх, ни вправо, ни вниз, ни влево, его движение останавливается.

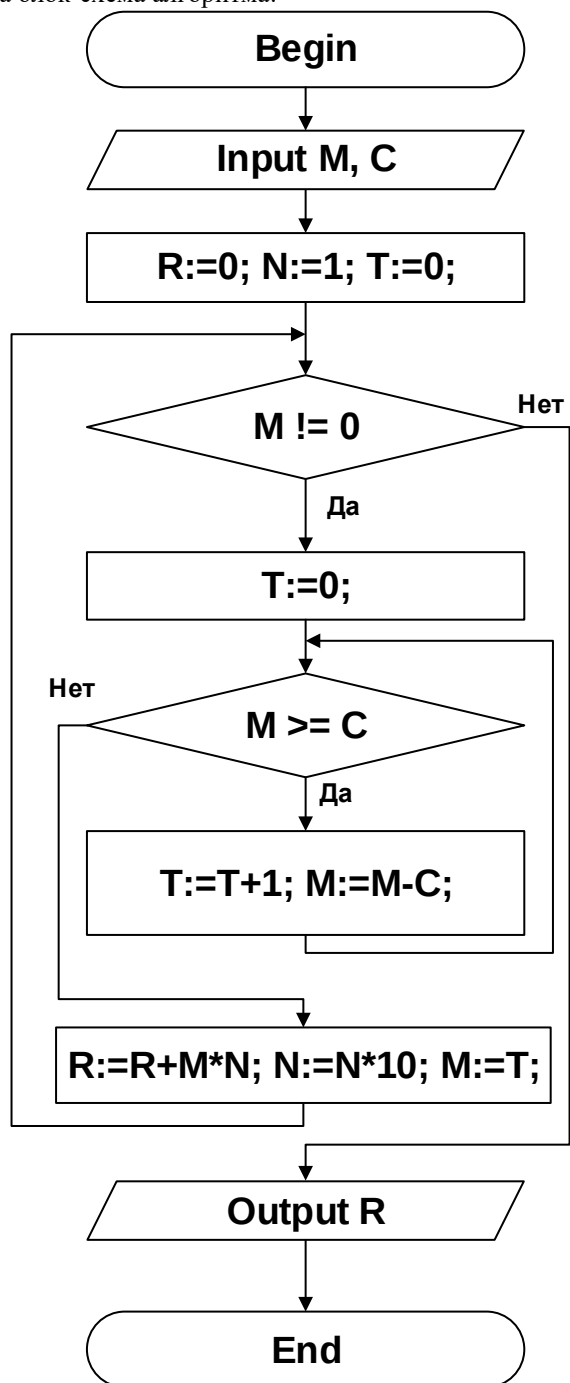
Сколько существует клеток, в которые можно изначально поместить робота так, что после остановки его движения окажется, что он посетил все клетки, которые на момент начала движения не были непроходимыми. В ответе укажите целое число.

Ответ: 7

9. Алгоритмизация и программирование. Блок-схема (1 балл)

[Вычитание]

Дана блок-схема алгоритма:



На вход подали два натуральных числа M и C . Известно, что на выходе получилось $R=30071$. Определите, какое значение M было на входе, если известно, что значение C на входе было равно 8. В ответе укажите целое число.

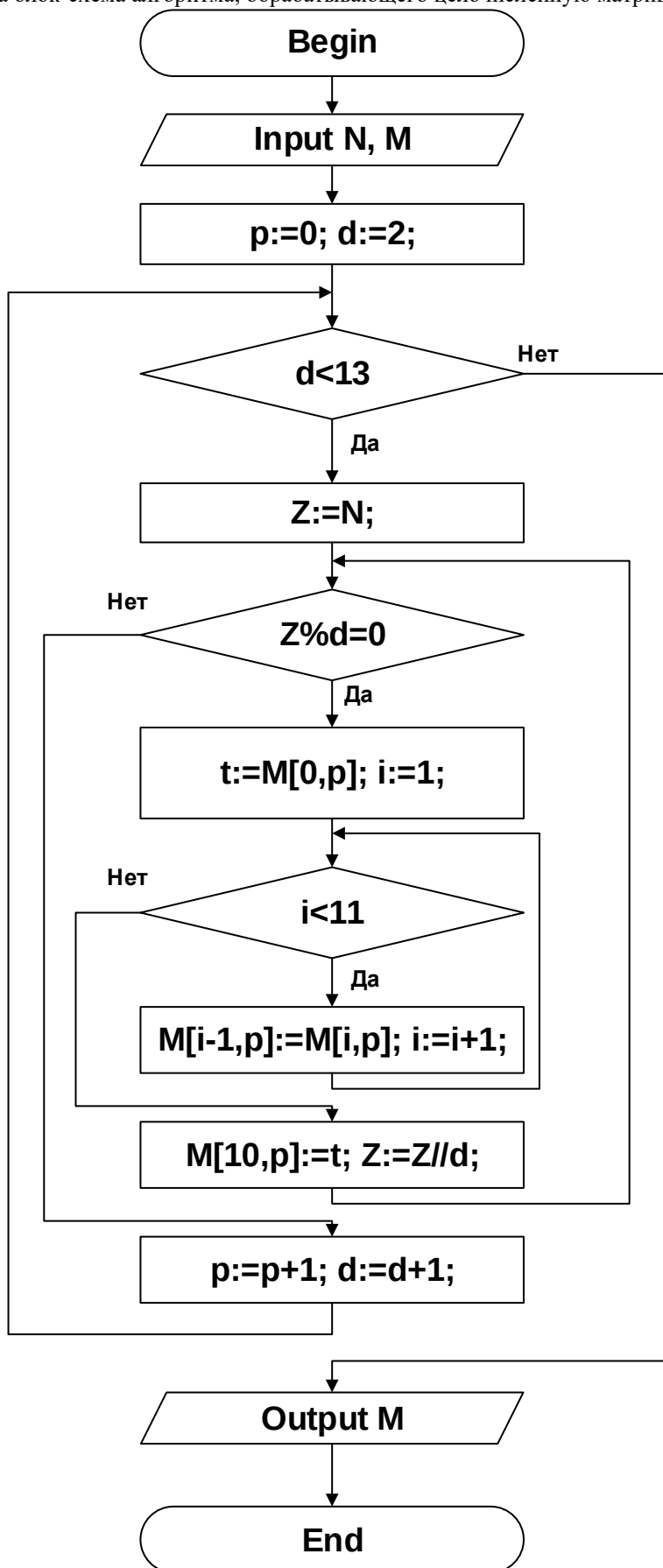
Примечание: оператор « $!=$ » означает «не равно».

Ответ: 12345

10. Алгоритмизация и программирование. Блок-схема, обратная задача (2 балла)

[Эквалайзер]

Дана блок-схема алгоритма, обрабатывающего целочисленную матрицу, размером 11 на 11 элементов:



На вход подали целое положительное число N и следующую матрицу M:

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

На выходе получили следующую матрицу M:

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	1	0	0	0
0	1	0	0	1	0	1	0	0	0	1	0
0	0	0	0	0	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0

При каком минимальном значении N это возможно? В ответе укажите целое положительное число. Если такого числа не существует, в ответе напишите NULL.

Примечания:

1. При обращении к элементам матрицы первый индекс означает номер строки, а второй – номер столбца. Нумерация производится от 0.
2. Операция $A\%B$ означает получение остатка от целочисленного деления A на B. Операция $A//B$ означает получение частного от целочисленного деления A на B.

Ответ: 504000

Отборочный этап. Второй тур (приведен один из вариантов заданий)

1. Электронные таблицы. Адресация ячеек и вычисления (3 балла)

[Снова 100x100]

Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D
1	1	=A1+1		
2	=A1+1	=A1+МИН(B\$1;\$A2)		
3				
4				
5				

Ячейку B1 скопировали во все ячейки диапазона C1:CV1. Ячейку A2 скопировали во все ячейки диапазона A3:A100. Ячейку B2 скопировали во все ячейки диапазона B2:##, где ## - адрес некоторой ячейки в диапазоне B2:CV100. Сколько существует различных адресов ячеек ##, таких, что в результате в диапазоне A1:## число 70 будет встречаться ровно 21 раз? В ответе укажите целое число.

Рекомендации к пояснению решения задачи.

В пояснении к решению задачи раскройте следующие вопросы:

1. Решали ли Вы задачу исключительно в электронных таблицах или использовали программирование и/или аналитические методы на каком-то этапе решения задачи?

2. Для части решения, использующего электронные таблицы, охарактеризуйте операции, которые Вы в них осуществили, назначение введенных в ячейки формул, и как это позволило получить ответ.

3. В случае программного решения, поясните, алгоритм, который Вы реализовали, включая назначение используемых переменных и структур данных, и как это позволило получить ответ.

4. Если в какой-то части решения Вы использовали аналитические методы решения, опишите, какие зависимости, закономерности и как Вы определили из условия или результатов вычислений в электронных таблицах, какие данные и формулы использовали для вычислений, и как это позволило получить ответ.

Рекомендуемое время пояснения решения 1-3 минуты.

Ответ: 62

2. Сортировка и фильтрация данных (2 балла)

[Маски файлов]

В ряде операционных систем для выделения подмножества из множества файлов в каталоге используются маски файлов. Для задания масок файлов используют следующие обозначения:

c – любой неспециальный символ, соответствует самому себе, не может быть звездочкой (*) или вопросительным знаком

(?).

? – ровно один неспециальный символ.

* – любое количество (в том числе 0) неспециальных символов.

Есть каталог с файлами и известно количество файлов, которые оказались выделены при использовании некоторых масок файлов:

Маска файла	Количество файлов
??*a?	100
cb*?d	45
*?***	300
c?*ae	70
?b*?d	200

В этом каталоге выделили все файлы, которые соответствуют хотя бы одной из двух масок:

ag*??

a?*f?

Какое максимальное количество файлов могло оказаться выделенными? В ответе укажите целое число.

Рекомендации к пояснению решения задачи.

В пояснении к решению задачи раскройте следующие вопросы:

1. Решали ли Вы задачу исключительно аналитически или использовали средства операционной системы или программирования?

2. В аналитической части решения опишите ход Ваших рассуждений, какие формулы Вы использовали для вычислений, и как это позволило получить ответ.

3. Если Вы в какой-то части решения использовали средства операционной системы, укажите её версию и как использование её средств позволило получить ответ.

Рекомендуемое время пояснения решения 1-3 минуты.

Ответ: 185

3. Мультимедиа технологии (2 балла)

[Шашечки]

Цветное изображение, пиксели которого закодированы с помощью цветовой модели RGB, может быть представлено в виде яркостных представлений отдельных каналов. На яркостном представлении канала белый цвет обозначает максимальную яркость этого канала в значении цвета пикселя изображения, а черный – нулевое значение яркости. Яркостные представления каналов расположены сверху вниз как красный канал, затем зеленый канал и затем синий канал. Исходное изображение содержало пиксели четырех различных цветов.

Обработка исходного изображения заключалась в следующем:

1. Исходное изображение перевели в цветовую модель HSB (Hue, Saturation, Brightness).
2. К цветовой координате Hue каждой точки прибавили некоторое значение ΔH . Поскольку цветовая координата Hue не может превышать 359, то если сумма оказывалась больше, из значения вычитали 360.
3. Изображение перевели обратно в цветовую модель RGB.

В результате таких обработок, отличающихся только значениями ΔH , получили 11 изображений, представленных в таблице:

Исходное изображение	Изображение №1	Изображение №2	Изображение №3	Изображение №4	Изображение №5
Изображение №6	Изображение №7	Изображение №8	Изображение №9	Изображение №10	Изображение №11

Определите и укажите в ответе через пробел два числа: сначала номер изображения, полученного в результате обработки со значением $\Delta H=60$, а затем номер изображения, полученного в результате обработки со значением $\Delta H=210$. Пример записи ответа: 1 2

Ответ: 11 3

4. Телекоммуникационные технологии (1 балл)

[Три узла]

В терминологии сетей TCP/IP маской сети называется двоичное 32-х разрядное число, определяющее, какая часть IP-адреса узла относится к адресу сети, а какая – к адресу самого узла в этой сети. При этом в маске сначала (в старших разрядах) стоят только единицы, а затем с некоторого разряда – только нули. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. Два узла считаются принадлежащими одной сети, если у них совпадают адреса сети.

Известно, что узлы с IP-адресами 192.168.215.91 и 192.168.151.91 принадлежат одной сети, а узел с IP-адресом 194.168.215.91 не принадлежит этой сети. Сколько существует различных масок сетей, при которых это возможно. В ответе укажите целое число.

Рекомендации к пояснению решения задачи.

В пояснении к решению задачи раскройте следующие вопросы:

1. Решали ли Вы задачу исключительно аналитически или использовали электронные таблицы и/или программирование на каком-то этапе решения задачи?

2. В аналитической части решения опишите, какие зависимости, закономерности и как Вы определили из условия, какие данные и формулы использовали для вычислений, и как это позволило получить ответ.

3. Если в какой-то части решения Вы использовали электронные таблицы, охарактеризуйте операции, которые Вы в них осуществили, назначение введенных в ячейки формул, и как это позволило получить ответ.

4. Если в какой-то части решения Вы использовали программирование, поясните, алгоритм, который Вы реализовали, включая назначение используемых переменных и структур данных, и как это позволило получить ответ.

Рекомендуемое время пояснения решения 1-3 минуты.

Ответ: 11

5. Операционные системы (3 балла)

[Кэш памяти]

Большинство современных операционных систем являются мультипрограммными и используют концепцию виртуальной памяти. Это означает, что в исполняемом коде каждой программы используются виртуальные адреса, отсчитываемые от 0 для каждой программы, а при обращении к ячейке памяти по виртуальному адресу операционная система осуществляет пересчет его в физический адрес. Эта процедура может отнимать значительное время, поэтому при организации вычислительного процесса важным элементом является кэш памяти.

Кэш памяти представляет собой ассоциативный массив, хранящий пары типа «ключ-значение», где в качестве ключа выступает виртуальный адрес, а в качестве значения вычисленный физический. Без использования кэша при каждом обращении программы к ячейке памяти приходилось бы осуществлять преобразование виртуального адреса в физический, что существенно замедляло бы вычисления. В случае использования кэша, для сначала производится поиск виртуального адреса в кэше. Если виртуальный адрес там нашелся, физический адрес берется из кэша. Такая ситуация называется «кэш попадание». Если виртуального адреса в кэше не оказалось, запускается процедура вычисления физического адреса. Такая ситуация называется «кэш промах». В случае кэш промаха в кэш помещается новая пара «виртуальный адрес – физический адрес», вытесняя ту пару, которая дольше всего не использовалась.

В зависимости от архитектуры вычислительной системы, кэш памяти может использоваться как для всех операций обращения, связанных с обращением к ячейке памяти так и только для конкретных, например, только операций чтения данных из ячейки памяти в регистр процессора.

Рассмотрим следующую модель вычислительной системы с кэшем памяти.

1. Программа представляет собой набор 16-битных инструкций, для удобства записанных в виде четырехразрядных шестнадцатеричных чисел. Старший разряд является номером исполняемой команды, а младшие 3 разряда – виртуальным адресом ячейки памяти, которая используется этой командой. В записи инструкции используются арабские цифры и строчные буквы из диапазона a-f.

1. Кэш памяти используется только для команд чтения ячейки памяти в регистр процессора, которым соответствует значение старшего шестнадцатеричного разряда инструкции «1». Например, инструкция 100a означает чтение данных из ячейки с виртуальным адресом 00a. Для любых других команд (инструкций, у которых старший шестнадцатеричный разряд отличается от 1) кэш памяти не используется.

2. Кэш памяти состоит из четырех ячеек, пронумерованных от 0 до 3 и изначально пустых. Каждая ячейка кроме пары «виртуальный адрес – физический адрес» хранит также счетчик. При выполнении каждой операции чтения выполняются следующие действия:

- Осуществляется попытка найти в кэше памяти виртуальный адрес, указанный в выполняемой инструкции.
- Если такой адрес найден в кэше (кэш попадание), счетчик для него устанавливается в 0.
- Если такой адрес не найден в кэше (кэш промах), то после вычисления физического адреса определяется пустая ячейка с минимальным номером, а если пустых ячеек нет, то определяется ячейка, значение счетчика которой максимально (если таких ячеек несколько, из них берется ячейка с минимальным номером). В эту ячейку помещается новая пара «виртуальный адрес – физический адрес», а счетчик этой ячейки устанавливается в 0.
- Независимо от того, получилось ли в результате выполнения инструкции кэш попадание или кэш промах, счетчики всех четырех ячеек увеличиваются на 1.

Вам дан файл с программой [ссылка на файл var1.txt]. Определите количество кэш попаданий, которое произошло при выполнении этой программы. В ответе запишите целое число.

Ответ: 303

6. Технологии программирования (2 балла)

[CSV]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

CSV (от англ. Comma-Separated Values — значения, разделённые запятыми) — это текстовый формат для представления таблиц. Каждая строка таблицы задается отдельной строкой текста, а различные столбцы отделены друг от друга запятыми.

В первой строке таблицы задаются заголовки столбцов, так же отделенные друг от друга запятыми. При этом значения в ячейках таблицы могут быть пустыми.

Вам необходимо вывести все заголовки полных столбцов. Столбец считается полным, если в каждой строке, кроме заголовочной, соответствующая ячейка не пуста.

Формат входных данных

Формат входных данных полностью соответствует формату CSV. Первая строка содержит заголовки.

Значения в каждой ячейке задаются строками из не более чем 2020 символов, состоящих из строчных букв латинского алфавита и цифр.

Гарантируется, что в таблице хотя бы две, но не более чем 100100 строк и хотя бы один, но не более 100100 столбцов. Также гарантируется, что заголовки столбцов различны и не пусты.

Формат выходных данных

В первой строке выведите число полных столбцов. Затем выведите все заголовки полных столбцов в лексикографическом порядке, каждый в отдельной строке.

Пример

Стандартный ввод	Стандартный вывод
surname,name,phone,test1,test2,test3,test4,extra,final,mark Ivanov,Pavel,,23,36,,45,47, Kuznetsov,Ivan,,78,88,,77,,45, Sidorov,Alexander,,90,80,,100,90,46, Popov,Mike,,11,0,,4,,,2 Bobov,Oleg,,,20,,30,,40,	4 name surname test2 test4

Замечание

Таблица из примера:

surname	name	phone	test1	test2	test3	test4	extra	final	mark
Ivanov	Pavel		23	36		45		47	
Kuznetsov	Ivan		78	88		77		45	
Sidorov	Alexander		90	80		100	90	46	
Popov	Mike		11	0		4			2
Bobov	Oleg			20		30		40	

7. Технологии программирования (4 балла)

[Шаблоны битовых матриц]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	4 секунды
Ограничение по памяти	256 мегабайт

Назовем шаблоном битовой матрицы размера $a \times b$ матрицу размера $a \times b$, которая может содержать только значения «0» и «?». При этом знак «?» означает любой символ.

Таким образом, например, шаблону

0	?
?	0

соответствуют четыре битовые матрицы:

0	1
1	0

0	1
0	0

0	0
1	0

0	0
0	0

Вам даны n шаблонов размера $a \times b$ и m битовых матриц такой же размерности. Необходимо для каждого шаблона определить сколько из данных матриц ему **не** соответствуют.

Обратите внимание на ограничения, если просто перебрать попарно матрицы и шаблоны, вы получите превышение ограничения по времени!

Формат входных данных

В первой строке заданы четыре целых числа a, b, n и m — размерности матриц, число шаблонов и число матриц, соответственно ($1 \leq a \cdot b \leq 12, 1 \leq n, m \leq 5 \cdot 10^4$).

В следующих $a \cdot n$ строках заданы шаблоны, каждый состоит из a строк по b символов. Каждый символ, соответственно, либо «0», либо «?».

Затем в $a \cdot m$ строках заданы матрицы, каждая матрица так же состоит из a строк по b символов. Каждый символ, соответственно, либо «1», либо «0».

Формат выходных данных

Выведите n чисел, разделенных пробелом — для каждого шаблона, сколько из данных матриц ему не соответствует.

Примеры:

Стандартный ввод	Стандартный вывод
2 2 3 4 0? ?0 00 00 ?0 ?? 10 01 11 11 00 00 10 11	3 3 1
2 3 1 2 ??? ??? 101 111 111 000	0

Задания для 9 и 10 класса

Заключительный этап, 10 класс (приведен один из вариантов заданий)

1. Кодирование информации. Системы счисления (2 балла)

[Много нулей]

Дано арифметическое выражение:

$$11_{16} \cdot (20_{16}^N + 1)^2 = X_2$$

Найдите минимальное целое положительное значение N , при котором в двоичной записи числа X будет более 1000 значащих нулей. Постройте двоичную запись числа X для найденного значения N и найдите в ней последовательность из идущих подряд нулей максимальной длины. Укажите в ответе длину найденной последовательности в виде числа в десятичной системе счисления.

Ответ: 501

Решение:

Обозначим за $\{0\}_K$ последовательность из K идущих подряд нулей. Тогда $20_{16}^N + 1 = 100000_2^N + 1$ можно записать как $1\{0\}_{(5N-1)}1$. Возведение этого числа в квадрат равносильно умножению его само на себя: $1\{0\}_{(5N-1)}1 \cdot 1\{0\}_{(5N-1)}1 = 1\{0\}_{(5N-1)}1 \cdot 1\{0\}_{(5N)} + 1\{0\}_{(5N-1)}1 = 1\{0\}_{(5N-1)}1\{0\}_{(5N)} + 1\{0\}_{(5N-1)}1 = 1\{0\}_{(5N-2)}1\{0\}_{(5N)}1$. Для получения значения X нужно еще умножить на $11_{16}=10001_2$. $1\{0\}_{(5N-2)}1\{0\}_{(5N)}1 \cdot 10001 = 1\{0\}_{(5N-2)}1\{0\}_{(5N)}10000 + 1\{0\}_{(5N-2)}1\{0\}_{(5N)}1$. Для любого $N > 1$ в результате получится: $10001\{0\}_{(5N-6)}10001\{0\}_{(5N-4)}1001$. Обратим внимание, что общее количество значащих нулей в записи будет $3+(5N-6)+3+(5N-4)+3 = 10N-1$, а максимальное количество идущих подряд нулей будет $5N-4$. Тогда минимальным N , при котором общее число нулей в записи числа X будет более 1000 является $N=101$. Следовательно, длина максимальной последовательности идущих подряд нулей будет равна $5 \cdot 101 - 4 = 501$.

2. Кодирование информации. Объем информации (2 балла)

[RLE]

Петя сохраняет в память текст. Известно, что алфавит текста составляет 32 символа. Петя использует равномерное кодирование и сохраняет в память коды символов, используя минимально возможное одинаковое для кодов всех символов количество бит.

Вася и Таня изучают кодирование длин серий, run-length encoding (RLE) и пытаются его применить к тексту Пети для того, чтобы он занимал меньше места в памяти.

В случае RLE кодирования весь текст представляется как расположенные друг за другом непересекающихся последовательности из одинаковых символов так, что символы в двух соседних последовательностях отличаются. При этом в памяти сохраняется для каждой последовательности два значения: длина последовательности и код повторяющегося в ней символа. Для записи кода символа используется минимальное возможное одинаковое для кодов всех символов количество бит. Для записи длины последовательности используется минимально возможное одинаковое для всех возможных значений этого параметра количество бит.

Вася обнаружил, что максимальная длина последовательности из одинаковых символов, которая встречается в тексте, равна M и решил, что в строке могут встречаться все возможные длины последовательностей, не превышающие M . Исходя из этого предположения, он определил количество бит, необходимое для записи длины последовательности и сохранив в память текст Пети с помощью кодирования RLE обнаружил, что он занимает ровно 208 байт памяти.

Также Вася обнаружил, что максимальная длина последовательности из одинаковых символов ровно в X раз меньше общего количества символов в тексте Пети.

Таня проанализировала текст более внимательно и обнаружила, что количество различных длин последовательностей, которые встречаются в тексте ровно в 16 раз меньше, чем длина максимальной последовательности. Исходя из этого, она определила другое количество бит, необходимое для записи длины последовательности и сохранив в память текст Пети с помощью кодирования RLE обнаружила, что он занимает ровно 144 байта памяти.

Определите значение X , если известно, что текст, сохраненный Петей, занимает на 12272 байта больше памяти, чем текст, сохраненный Васей. Если таких значений несколько, определите минимальное из них. В ответе укажите целое число.

Ответ: 78

Решение:

Поскольку текст, сохраненный Васей занимает 208 байт и это на 12272 байта меньше, чем текст, сохраненный Петей, объем памяти, который занимает текст, сохраненный Петей составляет $12272+208=12480$ байт.

Для сохранения кода одного символа при равномерном кодировании для алфавита из 32 символов потребуется $\log_2(32)=5$ бит.

Следовательно, количество символов в тексте равно $12480 \cdot 8 / 5 = 19968$. И это значение представимо как произведение искомой нами величины X на пока неизвестное значение M . То есть $X=19968/M$.

Найдем значение M . Для этого составим уравнения исходя из известным нам значений объемов памяти, занимаемых при кодировании RLE Васей и Таней.

Обратим внимание, что поскольку по условию требуется найти минимальное значение X , нам требуется найти максимальное значение M .

Пусть N – количество последовательностей, на которое разбивается текст Пети, тогда:

Вася: $N \cdot (\log_2(M) + \log_2(32)) = 208 \cdot 8$

Петя: $N \cdot (\log_2(M/16) + \log_2(32)) = 144 \cdot 8$

Поскольку мы ищем максимальное значение M , мы можем считать, что в уравнениях стоят именно значения $\log_2(M)$, а не их округления в большую сторону до ближайшего целого числа.

Вычтем из первого уравнения второе:

$$N \cdot (\log_2(M) + \log_2(32)) - N \cdot (\log_2(M/16) + \log_2(32)) = (208 - 144) \cdot 8$$

$$N * ((\log_2(M) + \log_2(32)) - (\log_2(M/16) + \log_2(32))) = 512$$

$$N * (\log_2(M) + \log_2(32)) - \log_2(M/16) - \log_2(32) = 512$$

$$N * (\log_2(M) - \log_2(M/16)) = 512$$

$$N * (\log_2(M / (M/16))) = 512$$

$$N * (\log_2(16)) = 512$$

$$N * 4 = 512$$

$$N = 128$$

Подставим найденное значение N в первое уравнение:

$$128 * (\log_2(M) + \log_2(32)) = 208 * 8$$

$$128 * (\log_2(M) + 5) = 1664$$

$$\log_2(M) + 5 = 1664 / 128$$

$$\log_2(M) = 8$$

$$M = 256$$

$$\text{Таким образом, } X = 19968 / 256 = 78$$

3. Основы логики (3 балла)

[Много чисел]

Логический преобразователь работает следующим образом:

1. На вход преобразователю подаётся целое неотрицательное число N, меньшее 2^{64}_{10} , записанное в шестнадцатеричной системе счисления. Запись числа переводится в двоичную систему счисления и разбивается на 16 групп по 4 разряда (при необходимости в начале записи дописываются незначащие нули).
2. Двигаясь справа налево по очереди берется каждая группа двоичных разрядов и представляется как последовательность из четырех логических переменных $X_0X_1X_2X_3$, так, что единичное значение двоичного разряда соответствует значению "истина", а нулевое - значению "ложь". Например, в числе 0...010111 первая такая последовательность будет $X_0 = \text{«ложь»}$, $X_1 = \text{«истина»}$, $X_2 = \text{«истина»}$, $X_3 = \text{«истина»}$.
3. Полученные последовательности логических переменных по очереди подставляются в логическое выражение:

$$(X_0 \vee X_1 \wedge X_2) \wedge (X_1 \vee X_2 \wedge X_3) \wedge (X_2 \vee X_3 \wedge X_0)$$
и вычисляется его значение.
4. Вычисленные значения преобразуются в двоичные разряды (также «истина» = 1, а «ложь» = 0) и образуют двоичную последовательность, заполняемую от младшего к старшему разряду.
5. Результатом работы преобразователя является шестнадцатиразрядное двоичное число (запись может содержать незначащие нули).

На вход преобразователю подали запись числа, равного FEDCBA9876543210₁₆ и получили некоторое число M. Сколько всего существует таких чисел, которые можно подать на вход преобразователю и получить в результате такое же число M? Поскольку число может получиться достаточно большое, посчитайте и укажите в ответе сумму его цифр при записи в десятичной системе счисления.

Ответ: 27

Решение:

Построим таблицу истинности для заданного в условии выражения. Это можно сделать как вручную, так и написав программу. Например, так:

```
N=int('FEDCBA9876543210',16)
```

```
tab=[]
```

```
for i in range(16):
```

```
    num=N%16
```

```
    st=[]
```

```
    for j in range(4):
```

```
        st.append(num%2)
```

```
        num//=2
```

```
    F=1 if (st[3]==1 or st[2]==1 and st[1]==1) and (st[2]==1 or st[1]==1 and st[0]==1) and (st[1]==1 or st[0]==1 and st[3]==1)
```

```
else 0
```

```
    tab.append([st[::-1],F])
```

```
    N//=16
```

```
for i in tab:
```

```
    print(i)
```

X_0	X_1	X_2	X_3	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0

1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Заметим, что в таблице 10 строк имеют ложное значение и 6 строк имеют истинное значение.

Обратим внимание, что преобразователь, разбивая двоичную запись на 16 групп по 4 разряда получает двоичные записи отдельных цифр шестнадцатеричного числа (при необходимости, дополненного незначащими нулями). Эти записи можно интерпретировать как значения логических переменных в строках таблицы истинности. Приведенное в условии число FEDCBA9876543210₁₆ содержит все возможные шестнадцатеричные цифры, расположенные по убыванию, следовательно, если перевести это число в двоичную систему счисления и начать рассматривать группы по 4 разряда с конца к началу записи, как указано в условии, мы получим все комбинации значений логических переменных, то есть все строки построенной выше таблицы истинности. Тогда число М получается из последнего столбца как 1110100011000000₂ = 59584₁₀.

Значение выражения для каждой строки таблицы истинности получается независимо и может быть истинно или ложно. Тогда, получается, что каждая цифра шестнадцатеричного числа даёт либо 0 либо 1 в очередном по порядку следования двоичном разряде числа М. А из таблицы истинности мы знаем какая цифра какое значение даёт и в каком разряде должно быть какое значение. Следовательно, любая цифра, которая даст значение 1 может быть на любой позиции, на которой должно получиться значение 1, а любая цифра, которая даст значение 0 может быть на любой позиции, на которой должно получиться значение 0. Теперь вспомним наблюдение, которое мы записали после таблицы истинности и выведем выражение для вычисления количества подходящих чисел: $10^{10} \cdot 6^6 = 46656000000000$. Сумма цифр в этом числе равна 27, что и является правильным ответом.

4. Алгоритмизация и программирование. Формальный исполнитель (1 балл)

[Лабиринт]

Дан лабиринт, размером 10 на 10 клеток:

```
* . * * * * * * *
* . . . . . . . *
* . * . * * . * .
* . * . . . . * .
* . * . * * . * .
* . * . * * . * .
* . * . . . . * .
* . * * * * * . *
* . . . . . . . *
* * * * * * * *
```

Каждая клетка содержит один из двух символов: «*» означает непреодолимое препятствие, а «.» свободная клетка. В одну из свободных клеток помещают шарик. Если ниже этой клетки есть еще свободные клетки, шарик падает, пока не наталкивается на непреодолимое препятствие. Далее лабиринт поворачивают следующим образом:

1. На 90 градусов по часовой стрелке
2. На 90 градусов по часовой стрелке
3. На 90 градусов по часовой стрелке
4. На 90 градусов против часовой стрелки.

После каждого поворота, если ниже той клетки, в которой находится шарик, есть еще свободные клетки, шарик падает, пока не наталкивается на непреодолимое препятствие. После этого можно осуществлять следующий поворот.

Сколько существует клеток, в которые можно исходно поместить шарик так, что в результате выполнения указанных поворотов (возможно до их завершения), он выкатится из лабиринта. В ответе укажите целое число.

Ответ: 43

Решение:

Можно перебрать точки вручную, но в этом случае легко допустить ошибку по невнимательности. Можно реализовать алгоритм в виде программы. Ниже приведен пример на языке Python. Программа избыточна, поскольку демонстрирует также траектории для всех найденных точек, что может быть удобно для анализа решения задания. Для нахождения только ответа на задание её можно упростить.

```
Path='RRRL'
ans=0
for x in range(10):
    for y in range(10):
        A=[]
        A.append(list("*.*****"))
```

```

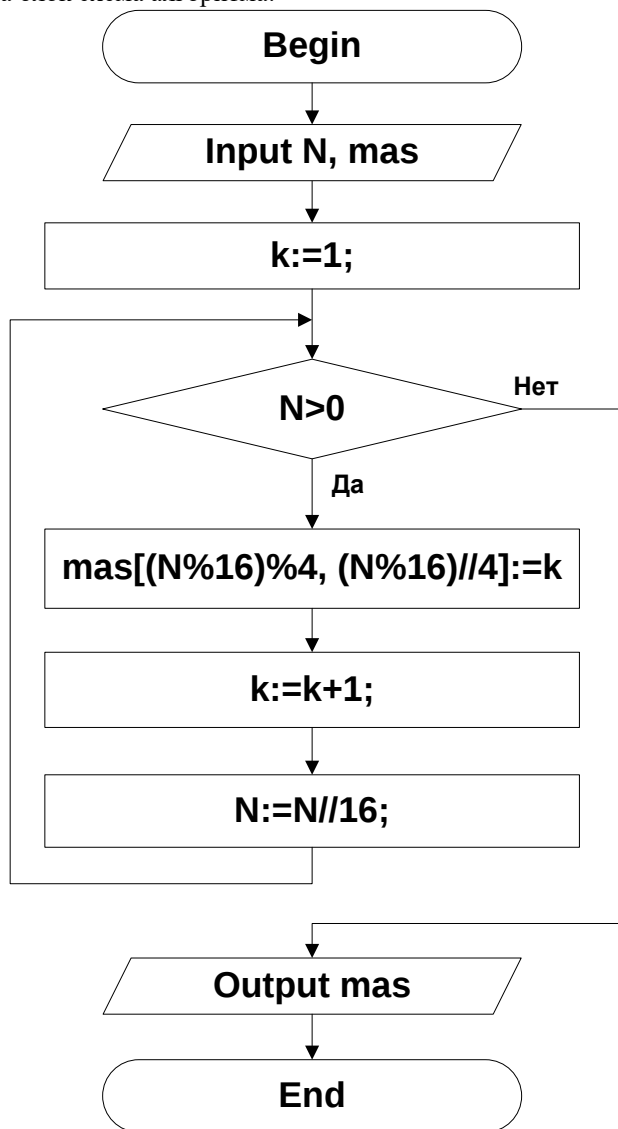
A.append(list("*.....*"))
A.append(list("*.*.*.*.*"))
A.append(list("*.*....*.*"))
A.append(list("*.*.*.*.*"))
A.append(list("*.*.*.*.*"))
A.append(list("*.*....*.*"))
A.append(list("*.******.*"))
A.append(list("*.....*"))
A.append(list("*****"))
if A[x][y]!='*':
    C=[x,y]
    A[C[0]][C[1]]='o'
    while A[C[0]+1][C[1]]!='*':
        C[0]+=1
        A[C[0]][C[1]]='o'
    D=0 #Направление: 0 - вниз, 1 - вправо, 2 - вверх, 3 - влево
    out=False
    for i in range(len(Path)):
        if not out:
            D=(D+1)%4 if Path[i]=='R' else (D+3)%4
            if D==0:
                while A[C[0]+1][C[1]]!='*':
                    C[0]+=1
                    A[C[0]][C[1]]='o'
            elif D==1:
                while A[C[0]][C[1]+1]!='*':
                    C[1]+=1
                    A[C[0]][C[1]]='o'
            elif D==2:
                while A[C[0]-1][C[1]]!='*':
                    C[0]-=1
                    A[C[0]][C[1]]='o'
                if C==[0,1]:
                    print('Finish from', x, y)
                    for i in A:
                        print("".join(i))
                    out=True
                    ans+=1
                    break
            else:
                while A[C[0]][C[1]-1]!='*':
                    C[1]-=1
                    A[C[0]][C[1]]='o'
print(ans)

```

5. Алгоритмизация и программирование. Анализ алгоритма, заданного в виде блок-схемы (2 балла)

[4*4=16]

Дана блок-схема алгоритма:



На вход алгоритма подали целое положительное число N и целочисленную матрицу mas , размером 4 на 4 элемента, заполненную нулями. На выходе получили следующую матрицу mas :

14	9	4	7
13	5	2	1
6	10	8	16
15	12	3	11

Определите минимальное число, N для которого это возможно. Представьте запись числа N в двоичной системе счисления, сложите те цифры этой записи, номер которых (нумеруя цифры с 0 от младшего разряда к старшему) делится нацело на 3 и запишите результат в десятичной системе счисления. Например, если $N=10101001_2$, ответом будет 2.

Примечания. Операция $A\%B$ вычисляет остаток от целочисленного деления A на B ; операция $A//B$ вычисляет частное от целочисленного деления A на B . При обращении к элементам матрицы первый индекс – номер строки, а второй – номер столбца, нумерация с (0,0).

Ответ: 12

Решение:

Заметим, что на каждом шаге цикла происходит обращение к значению $N\%16$, а потом деление $N//16$. Это не что иное как разбор на отдельные цифры шестнадцатеричной записи числа N . То есть, на каждом шаге цикла берется цифра с номером k (считая справа налево от 1) в записи числа в шестнадцатеричной системе счисления. Вспомним, что любая шестнадцатеричная цифра может быть представлена как пара цифр для записи в четверичной системе счисления, например $D_{16}=31_4$. Тогда $(N\%16)\%4$ – это младшая (вторая) цифра в такой паре, а $(N\%16)//4$ – старшая (первая) цифра в такой паре. Тогда операция $mas[(N\%16)\%4, (N\%16)//4]:=k$ означает, что в элемент матрицы, номер строки которого соответствует младшей цифре такой пары, а номер столбца – старшей цифры записывается номер соответствующей шестнадцатеричной цифры. Значит, мы можем брать последовательно элементы матрицы со значениями от 1 до 16, формировать для них пары (номер столбца, номер строки) и записывать такие пары справа налево (заметим, что нумерация идет от нуля):

32030001133312102230021120232131

Пользуясь тем, что любая четверичная цифра может быть представлена независимо как две двоичных, переведем в двоичную систему счисления и выделим разряды, номера которых (при нумерации с 0 от младшего к старшему разряду) делятся нацело на 3:

11100011000000101111110110010010000000101000101110011101

Посчитаем количество выделенных единиц – их будет ровно 12, что и является ответом на задание.

6. Телекоммуникационные технологии (2 балла).

[Три порта]

Петя моделирует работу прототипа коммутатора. У этого прототипа есть три порта. Два порта принимают данные из двух сетей со скоростью 5 КБайт в секунду каждый. Третий порт отдает данные в третью сеть со скоростью 10 КБайт в секунду. Данные из первой сети приходят пакетами по 40 КБайт, причем первый пакет начинает приниматься в начальный момент времени и далее новый пакет начинает приниматься по истечении каждых 10 секунд. Данные из второй сети приходят пакетами по 30 КБайт, причем первый пакет начинает приниматься по истечении 4 секунд от начального момента времени и далее новый пакет начинает приниматься по истечении каждых 8 секунд.

Прототип коммутатора работает следующим образом. Поступающий пакет по мере передачи данных записывается в буферную память. Если одновременно принимаются пакеты в оба порта их запись в буферную память происходит одновременно и независимо. Как только некоторый пакет целиком принят, он становится готов к передаче через выходной порт. Передача данных через выходной порт происходит исходно полученными пакетами по 40 и по 30 КБайт соответственно, причем, если в некоторый момент времени есть возможность начать передать пакеты обоих размеров, сначала передается пакет большего размера. В один момент времени через выходной порт может передаваться только один пакет и пакет всегда передается целиком. Передача пакета через выходной порт начинается сразу, как только появляется готовый пакет, если в этот момент не передается другой пакет или сразу после завершения передачи предыдущего пакета. Как только очередной пакет передан, он удаляется из буфера. Если в один момент времени нужно удалить переданный пакет и записать очередную порцию данных принимаемого пакета, сначала происходит удаление. Если вся память в буфере занята, но происходит попытка принять данные, происходит переполнение буфера и аварийная остановка коммутатора. Определите минимальный размер буфера в КБайтах, при котором в процессе работы данной модели никогда не произойдет аварийная остановка. В ответе укажите целое число.

Ответ: 90

Решение:

Вручную или с помощью электронной таблицы построим модель работы прототипа коммутатора. Одна строка означает состояние модели по истечении очередной секунды. В столбцах «Порт 1» и «Порт 2» отображается объем данных в КБайт, полученных через этот порт к очередному моменту времени. В столбце «Выход» показаны в виде отрицательных чисел данные, удаленные из буфера к очередному моменту времени. В столбце «Буфер» считается объем данных, хранимых в буфере в указанный момент. Желтые клетки означают секунду, на которой заканчивается приём очередного пакета. Со следующей секунды возможна его передача. Зеленые клетки означают секунды, на которых передаётся в третью сеть пакет, размером, 40 КБайт, а голубые – размером 30 КБайт.

Секунда	Порт 1	Порт 2	Выход	Буфер
1	5			5
2	10			10
3	15			15
4	20			20
5	25	5		30
6	30	10		40
7	35	15		50
8	40	20		60
9	40	25		65
10	40	30		70
11	45	30		75
12	50	30		80
13	55	35	-40	50
14	60	40	-40	60
15	65	45	-40	70
16	70	50	-70	50
17	75	55	-70	60
18	80	60	-70	70
19	80	60	-70	70
20	80	60	-70	70
21	85	65	-70	80

22	90	70	-70	90
23	95	75	-110	60
24	100	80	-110	70
25	105	85	-110	80
26	110	90	-140	60
27	115	90	-140	65
28	120	90	-140	70
29	120	95	-140	75

Обратим внимание, что по истечении 22-ой секунды от начального момента времени в буфере занято 90 КБайт. При этом, уже в начале следующей секунды из буфера будет удалено 40 КБайт, а поступит 10, причем по условию удаление будет предшествовать заполнению. Легко заметить, что достижение занятости в буфере 90 КБайт происходит в худшем случае, когда одновременно завершён на 18-й секунде прием сразу двух пакетов. В этом случае будет происходить накопление данных в буфере во время передачи большего пакета при наличии в памяти полного пакета меньшего размера. В принципе, можно продолжить таблицу и убедиться, что вторая такая же ситуация с одновременным завершением приёма двух пакетов произойдет по истечении 56-й секунды и, следовательно, занятость буфера в 90 КБайт следующий раз будет достигнута на 60-й секунде. Следовательно, процесс изменения памяти буфера будет цикличен и значение 90 КБайт никогда не будет превышена. Значит это и есть максимальный объем буфера, при котором будет неограниченная во времени безаварийная работа.

7. Технологии обработки информации в электронных таблицах, сортировки и фильтрации данных (1 балл)

[Остатки]

Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D	E	F	G	H	I	J
1		2	3	5	6	10	15	30		
2		=СЧЁТЕСЛИ(B3:B102;"0")								
3		=ОСТАТ(\$A3;\$B\$1)							=СЧЁТЕСЛИ(B3:H3;"0")	
4										

Ячейку B2 скопировали во все ячейки диапазона C2:I2. Ячейку B3 скопировали во все ячейки диапазона B3:H102. Ячейку I3 скопировали во все ячейки диапазона I4:I102. Диапазон A3:A102 заполнили некоторым набором целых положительных чисел.

Известны значения некоторых ячеек, получившиеся после этого:

	A	B	C	D	E	F	G	H	I	J
1		2	3	5	6	10	15	30		
2		50	33	20	17	10	7	4		
3										
4										
5										

Определите значение в ячейке I2. В ответе укажите целое число.

Ответ: 27

Решение:

Обратим внимание на числа в диапазоне B1:H1. Сначала идут три простых числа (2, 3, 5), затем три пары их возможных произведений ($2*3=6$, $2*5=10$, $3*5=15$) и, наконец, произведение всех трех чисел ($2*3*5=30$).

Диапазон B3:B102 заполнен формулой =ОСТАТ(\$A3;\$B\$1). То есть, в столбце B этого диапазона будут остатки от деления чисел из диапазона A3:A102 на 2, в столбце C – остатки от деления на 3, в столбце D – остатки от деления на 5 и т.д.

Тогда в ячейках диапазона B2:H2 будут числа, означающие, сколько чисел из диапазона A3:A102 делятся на 2, сколько на 3, сколько на 5, сколько на 6, сколько на 10, сколько на 15 и сколько на 30. Обратим внимание, что эти числа можно интерпретировать как мощности трех пересекающихся множеств: множество чисел из диапазона A3:A102, которые делятся нацело на 2, множество чисел этого же диапазона, которые делятся нацело на 3 и множество чисел из этого диапазона, которые делятся нацело на 5. Формулы из диапазона I3:I102 подсчитывают для каждого числа количество делителей из набора {2, 3, 5}. Значение 0 в этом столбце будет только у тех чисел, которые не делятся ни на 2, ни на 3, ни на 5. Тогда в ячейке I2, значение которой нам нужно найти, будет количество таких чисел из диапазона A3:A102, которые не делятся ни на 2, ни на 3, ни на 5.

Посчитаем мощность объединения множеств, чисел из диапазона A3:A102, которые делятся на 2, 3 или 5. Поскольку мы знаем про все пересечения отдельных множеств, воспользуемся формулой включений-исключений и получим $M=50+33+20-17-10-7+4=73$. Всего в диапазоне A3:A102 ровно 100 чисел. Следовательно, $100-73=27$ из них не делятся ни на 2, ни на 3, ни на 5. Значит, именно это число в ячейке I2.

8. Технологии программирования (2 балла)

[Биржа]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

В данной задаче вам требуется реализовать подсчет прибыли при торговле акциями.

Торги проходят в течении n дней, всего на рынке представлены акции m компаний, цены акций компании фиксированы в течение одного дня. Сделки бывают двух типов:

Купить x акций компании *comp*

Продать все акции компании *comp*

За каждую сделку надо заплатить 1% комиссии. Например, если купить 10 акций по 300 рублей, то суммарно заплатить придется 3030 рублей. Если же продавать 10 акций стоимостью 300 рублей каждая, то за них можно получить 2970 рублей.

Прибылью с продажи будем считать разность полученных при продаже денег и суммарно потраченных денег при покупках. Например, если 10 акций были куплены по 300 рублей, а затем еще 5 акций были куплены по 400 рублей, то в случае продажи по стоимости 500 прибыль составит: $15 \cdot 500 - 0.99 \cdot (10 \cdot 300 + 5 \cdot 400) = 7425 - (3030 + 2020) = 2375$ рублей. При этом акции могут быть проданы в убыток (за меньшую стоимость, чем были куплены), тогда прибыль с продажи будем считать отрицательной.

Суммарной прибылью в момент времени будем считать сумму всех прибылей с продаж, которые произошли до этого момента. Обратите внимание, что деньги, потраченные к этому моменту на покупку акций, которые еще не были проданы, не учитываются. Также будем считать, что у нас неограниченный бюджет для покупки акций, но рассматривать будем только прибыль. Число акций для покупки/продажи для каждой компании на рынке также не ограничено.

Вам даны k событий покупки/продажи. Необходимо найти минимальную суммарную прибыль среди всех моментов времени.

Формат входных данных

В первой строке входных данных содержится одно целое число t — число тестовых наборов ($1 \leq t \leq 30$).

Затем следуют t тестовых наборов. Каждый тестовый набор описывается следующим образом:

В первой строке тестового набора содержатся три целых числа n , m и k — число дней, в которые проходят торги, число компаний на рынке и число событий, соответственно ($1 \leq n, m \leq 100$, $1 \leq k \leq 1000$).

В следующих m строках записаны названия компаний и n чисел — стоимости акций компании в рублях в каждый из дней торгов. Названия компаний состоят из не более чем 10 строчных букв латинского алфавита и попарно различны. Стоимости акций — целые числа в диапазоне от 1 до 10^5 включительно.

В следующих k строках заданы события покупки/продажи в хронологическом порядке. Событие покупки задается в формате **<день> buy <число акций> <название компании>**, а событие продажи задается в формате **<день> sell <название компании>**. При этом **<день>** — целое число от 1 до n , а **<число акций>** — целое число от 1 до 1000. Гарантируется, что все события следуют в порядке неубывания дней и корректны, а именно нет продаж некупленных акций и покупок акций, которых нет на рынке.

Формат выходных данных

Для каждого тестового набора выведите в отдельной строке минимальную прибыль среди всех моментов времени, с относительной или абсолютной погрешностью не более 10^{-4} .

Пример

Стандартный ввод	Стандартный вывод
4	0
3 1 3	-11.11
comp 300 400 500	-2080.0
1 buy 10 comp	0
2 buy 5 comp	
3 sell comp	
3 2 4	
gazp 100 111 300	
yndx 1000 1100 1111	
1 buy 10 gazp	
2 buy 1 yndx	
3 sell yndx	
3 sell gazp	
3 1 3	
comp 300 400 200	
1 buy 10 comp	
2 buy 5 comp	
3 sell comp	
2 2 3	
bdn 100 100	
nik 1 100	
1 buy 300 bdn	
1 buy 10 nik	
2 sell nik	

Замечание

В первом тестовом наборе изначально до продаж суммарная прибыль равна 0, после первой продаже суммарная прибыль становится 2375 (случай разобран в примере).

Во втором тестовом наборе промежуточные прибыли равны 0, −11.11 (акция продана дороже, но комиссия больше разницы) и 1948.89.

В третьем тестовом наборе промежуточные прибыли равны 0 и −2080.

В четвертом тестовом наборе промежуточные прибыли равны 0 и 979.9, деньги, потраченные на непроданные акции, не учитываются.

Решение

Для решения этой задачи необходимо написать ровно то, что просят в условии. Наиболее удобно это можно сделать, если воспользоваться ассоциативным массивом, он же `std::map` в C++ или `dict` в Python.

Тогда, обрабатывая события, будем для каждой компании поддерживать сколько денег мы потратили на покупку акций и сколько у нас акций этой компании есть в текущий момент. Также надо не забыть домножить на 1.01, чтобы учесть комиссию.

При обработке события продажи будем умножать количество имеющихся акций на текущую цену и на 0.99, потому что комиссия будет идти уже в минус. После чего добавит к текущей прибыли, разность полученных денег потраченных денег, которые мы как раз храним в ассоциативном массиве.

Соответственно далее остается только найти минимум/максимум «промежуточной» прибыли.

С примерами авторских решений можно ознакомиться в папке `solutions` в архиве задачи.

9. Технологии программирования (4 балла)

[Трансформация массива]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

Вам дано t пар массивов a_i и b_i равной длины.

За одну операцию модификации можно:

Поменять местами любые два элемента массива a_i , но каждый элемент массива может участвовать не более чем в одном обмене.

Прибавить к любому элементу массива a_i единицу. Данную операцию можно применять неограниченное число раз к любому элементу массива.

Для каждой пары массивов найдите минимальное число операций, которые необходимо применить к массиву a_i , чтобы получить массив b_i , или определите, что это невозможно.

Формат входных данных

В первой строке дано число t — число пар массивов ($1 \leq t \leq 40$).

В следующих $3t$ строках содержатся описания пар массивов. Каждая пара описывается тремя строками.

В первой из них дано число n_i — количество элементов в каждом массиве i -й пары ($1 \leq n_i \leq 10$). Во второй строке заданы n_i чисел $a_{i,j}$ — элементы массива a_i ($1 \leq a_{i,j} \leq 1000$). В третьей строке заданы n_i чисел $b_{i,j}$ — элементы массива b_i ($1 \leq b_{i,j} \leq 1000$).

Гарантируется, что сумма n_i по всем тестовым наборам не превосходит 150.

Формат выходных данных

Для каждой пары массивов выведите одно число — минимальное число операций, которые необходимо применить к массиву a_i , чтобы получить массив $b_{i,j}$, или −1, если для данной пары это невозможно.

Пример

Стандартный ввод	Стандартный вывод
6	-1
1	-1
2	1
1	-1
4	7
1 2 3 4	-1
2 4 1 2	
1	
1	
2	
5	
1 2 3 4 5	
1 5 1 1 1	
5	
1 2 3 4 5	
5 1 5 5 5	
3	
1 2 3	
3 1 2	

Решение

Данную задачу можно решать двумя способами: динамическим программированием или жадным подходом.

Для начала заметим, что если элемент массива a_i не участвует в обмене, то мы однозначно знаем сколько нужно операций, чтобы получить соответствующий элемент b_i (или что это невозможно). Также заметим, что после конкретного обмена мы также знаем сколько нужно операций изменения элемента.

Тогда, если решать динамическим программированием, можно сказать $dp[mask]$ — минимальное число операций, чтобы все элементы массива a_i с индексами соответствующими битовой маске $mask$, стали соответствующими элементами массива b_i . Решение очень похоже на поиск паросочетания минимального веса, в целом, тут и правда спрятан граф.

Для жадного решения нужно было доказать/поверить в то, что обмены довольно дешевые и их всегда выгодно делать, а затем перебирая элементы массива a_i в порядке возрастания находить оптимальную пару для обмена.

С примерами авторских решений можно ознакомиться в папке `solutions` в архиве задачи.

Заключительный этап, 9 класс (приведен один из вариантов заданий)

1. Кодирование информации. Системы счисления (2 балла)

[Много нулей]

Дано арифметическое выражение:

$$11_{16} \cdot (20_{16}^N + 1)^2 = X_2$$

Найдите минимальное целое положительное значение N, при котором в двоичной записи числа X будет более 1000 значащих нулей. Постройте двоичную запись числа X для найденного значения N и найдите в ней последовательность из идущих подряд нулей максимальной длины. Укажите в ответе длину найденной последовательности в виде числа в десятичной системе счисления.

Ответ: 501

Решение:

Обозначим за $\{0\}_K$ последовательность из K идущих подряд нулей. Тогда $20_{16}^N + 1 = 100000_2^N + 1$ можно записать как $1\{0\}_{(5N-1)}1$. Возведение этого числа в квадрат равносильно умножению его само на себя: $1\{0\}_{(5N-1)}1 * 1\{0\}_{(5N-1)}1 = 1\{0\}_{(5N-1)}1 * 1\{0\}_{(5N)} + 1\{0\}_{(5N-1)}1 = 1\{0\}_{(5N-1)}1\{0\}_{(5N)} + 1\{0\}_{(5N-1)}1 = 1\{0\}_{(5N-2)}1\{0\}_{(5N)}1$. Для получения значения X нужно еще умножить на $11_{16} = 10001_2$. $1\{0\}_{(5N-2)}1\{0\}_{(5N)}1 * 10001 = 1\{0\}_{(5N-2)}1\{0\}_{(5N)}10000 + 1\{0\}_{(5N-2)}1\{0\}_{(5N)}1$. Для любого $N > 1$ в результате получится: $10001\{0\}_{(5N-6)}10001\{0\}_{(5N-4)}1001$. Обратим внимание, что общее количество значащих нулей в записи будет $3 + (5N-6) + 3 + (5N-4) + 3 = 10N-1$, а максимальное количество идущих подряд нулей будет $5N-4$. Тогда минимальным N, при котором общее число нулей в записи числа X будет более 1000 является $N=101$. Следовательно, длина максимальной последовательности идущих подряд нулей будет равна $5*101-4=501$

2. Кодирование информации. Объем информации (1 балл)

[Моноширинный шрифт]

Петя пытается переписать прошивку детского планшета для рисования, чтобы использовать его как электронную книгу. Детский планшет умеет выводить черно-белые изображения (каждый пиксель или черный, или белый), размером 640 на 480 пикселей. Петя написал программу, которая формирует из текста отдельные страницы книги так, что на одной странице 60 строк по 80 символов, преобразует такую страницу в изображение, требуемого для планшета размера и для каждой страницы сохраняет в памяти коды отдельных пикселей получившегося изображения, используя один бит на код пикселя (метод А). Петя быстро понял, что при таком подходе в память планшета помещается очень мало страниц. Тогда он задумался, что в тексте, который он хочет выводить, встречается не так уж много различных символов. Он решил использовать моноширинный шрифт. В этом случае все изображение разбивается на непересекающиеся знакоместа, размером 8 на 8 пикселей и каждый символ отображается в своём знакоместе. Петя сохранил в памяти планшета изображения знакомест для всех возможных символов и теперь для каждой страницы хранит только коды символов для всех знакомест страницы (метод Б), а уже программа на планшете формирует из изображений отдельных букв окончательную картинку и выводит её. Коды хранятся для всех символов: даже, если в некотором знакоместе не выводится символ, хранится код пробела. Для хранения каждого кода символа Петя отводит в памяти минимально возможное, одинаковое для кодов всех символов количество бит. Петя обнаружил, что в случае использования метода Б на одну страницу тратится на 34800 байт меньше памяти, чем при использовании метода А. Определите максимальное количество символов в алфавите текста, при котором это возможно.

Ответ: 64

Решение:

Определим, сколько памяти в байтах потребуется для хранения страницы по методу А. Это будет $(640*480*1)/8=38400$ байт. Теперь запишем формулу для вычисления объема памяти, требуемой для хранения страниц по методу Б: $60*80*\log_2(X)$. При этом значение $\log_2(X)$ округляется до ближайшего большего целого числа. Значит, поскольку нас спрашивают про максимальное значение, можем считать, что нас интересует такое значение X, которое будет целой степенью 2, поскольку тогда его не нужно будет округлять вверх. Также отметим, что формула для определения информационного объема для страницы, сохраненной по методу Б, даёт ответ в битах. Тогда можно построить соотношение:

$$38400 - 60*80*\log_2(X)/8 = 34800$$

Получается, что $\log_2(X) = (38400 - 34800)*8/(60*80) = 6$. Тогда максимальное значение $X = 2^6 = 64$

3. Основы логики (3 балла)

[Из чего это следует?]

Две логические функции четырех переменных будут считаться не эквивалентными, если существует хотя бы одна комбинация значений переменных A, B, C и D, для которых логические функции принимают различные значения.

Сколько существует не эквивалентных друг другу логических функций от четырех переменных $F(A, B, C, D)$ таких, что будут одновременно истинны два логических выражения:

$$F(A, B, C, D) \rightarrow ((A \wedge \bar{B}) \leftrightarrow (C \wedge \bar{D})) = \text{Истина}$$

$$F(A, B, C, D) \rightarrow ((\bar{A} \wedge B) \leftrightarrow (\bar{C} \wedge D)) = \text{Истина}$$

В ответе укажите целое число.

Ответ: 64

Решение:

Вручную или с использованием программирования построим таблицы истинности для правых частей обеих импликаций (ноль означает ложное значение, а единица – истинное):

A	B	C	D	$((A \wedge \bar{B}) \leftrightarrow (C \wedge \bar{D}))$	$((\bar{A} \wedge B) \leftrightarrow (\bar{C} \wedge D))$
0	0	0	0	1	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	1	1	1
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	1	1

Обратим внимание, что, если для некоторой комбинации значений A, B, C, D правая часть хотя бы одной импликации принимает ложное значение, функция $F(A, B, C, D)$ также должна принимать ложное значение. В противном случае соответствующая импликация не сможет быть истинной. А вот для комбинаций значений A, B, C, D, для которых правая часть обеих импликаций истинна – функция $F(A, B, C, D)$ может принимать любое значение. Обозначим его вопросительным знаком и построим столбец с возможными значениями функции $F(A, B, C, D)$:

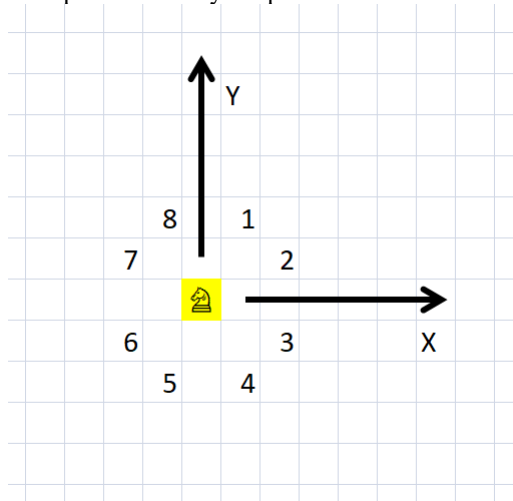
A	B	C	D	$((A \wedge \bar{B}) \leftrightarrow (C \wedge \bar{D}))$	$((\bar{A} \wedge B) \leftrightarrow (\bar{C} \wedge D))$	$F(A, B, C, D)$
0	0	0	0	1	1	?
0	0	0	1	1	0	0
0	0	1	0	0	1	0
0	0	1	1	1	1	?
0	1	0	0	1	0	0
0	1	0	1	1	1	?
0	1	1	0	0	0	0
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	0	0
1	0	1	0	1	1	?
1	0	1	1	0	1	0
1	1	0	0	1	1	?
1	1	0	1	1	0	0
1	1	1	0	0	1	0
1	1	1	1	1	1	?

Заметим, что функция может принимать любое значения для шести комбинаций переменных A,B,C,D. Следовательно, существует $2^6=64$ таких не эквивалентных функций.

4. Алгоритмизация и программирование. Формальные исполнители (1 балл)

[Ход конём]

Шахматный конь находится в центре клетчатого поля в клетке с координатами (0,0). Ось X направлена слева направо. Ось Y направлена снизу вверх.



Для управления конём есть 8 команд. На рисунке показаны клетки, в которые переместится конь из желтой клетки при выполнении команды с соответствующим номером.

Конь выполнил программу из 55 команд, про которую известно следующее:

1. Команд 1 в два раза больше, чем команд 8.
2. Команд 2 на 1 больше, чем команд 3.
3. Команд 3 на 2 больше, чем команд 8.
4. Команд 4 на 3 меньше, чем команд 2
5. Команд 5 на 2 больше, чем команд 6
6. Команд 6 на 2 меньше, чем команд 2
7. Команд 7 на 2 меньше, чем команд 2

Определите координаты клетки, в которой оказался конь, выполнив программу. В ответе укажите через пробел два числа: сначала координату X, а затем координату Y.

Ответ: 8 5

Решение:

Обозначим количество некоторых команд за X, например, количество команд 4. Тогда из условия 4 получим, что команд 2 будет $(X+3)$. По условию 2 получим, что команд 3 будет $(X+2)$. По условию 6 получим, что команд 6 будет $(X+1)$. По условию 7 получим, что команд 7 будет $(X+2)$. По условию 5 получим, что команд 5 будет $(X+3)$. По условию 3 получим, что команд 8 будет X. Наконец, по условию 1 получим, что команд 1 будет $X*2$.

Эти данные позволяют нам составить уравнение:

$$X*2 + (X+3) + (X+2) + X + (X+3) + (X+1) + (X+1) + X = 55.$$

Упростив, получим $9*X = 45$, $X=5$.

Составим таблицу с количеством команд и смещением по каждой координате:

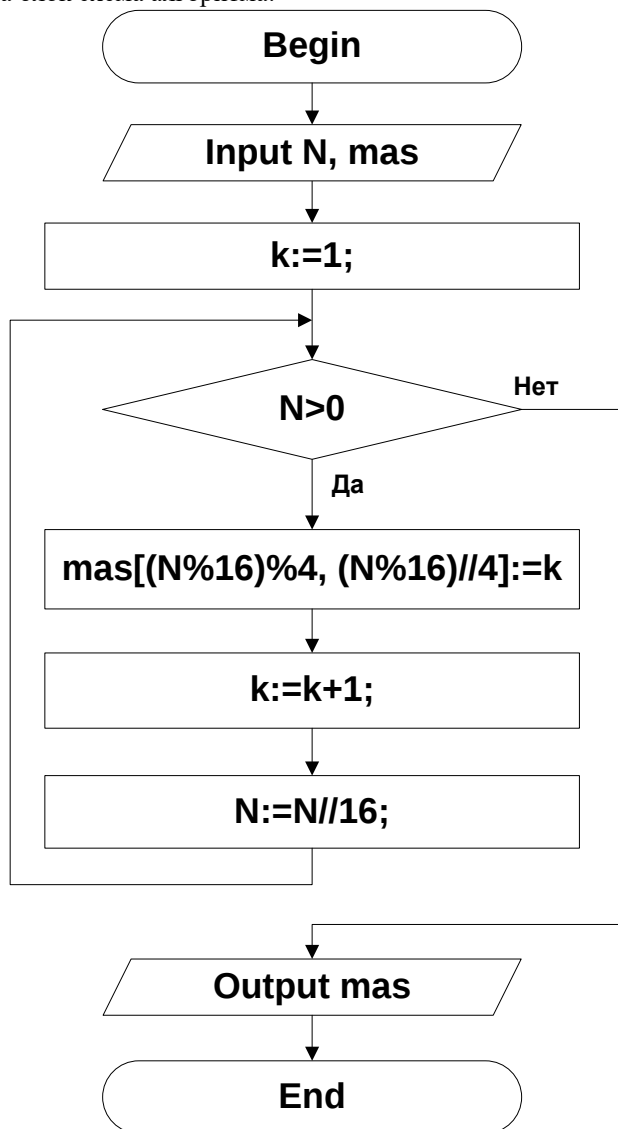
Команда	Количество команд		Смещение по X	Смещение по Y
1	$X*2$	10	10	20
2	$X+3$	8	16	8
3	$X+2$	7	14	-7
4	X	5	5	-10
5	$X+3$	8	-8	-16
6	$X+1$	6	-12	-6
7	$X+1$	6	-12	6
8	X	5	-5	10
Итого		55	8	5

Как видно, произошло смещение на 8 клеток по оси X и на 5 клеток по оси Y, то есть, начав в точке с координатами (0,0) конь, выполнив программу, оказался в клетке с координатами (8,5), то есть ответ «8 5».

5. Алгоритмизация и программирование. Анализ алгоритма, заданного в виде блок-схемы (3 балла)

[4*4=16]

Дана блок-схема алгоритма:



На вход алгоритма подали целое положительное число N и целочисленную матрицу mas , размером 4 на 4 элемента, заполненную нулями. На выходе получили следующую матрицу mas :

14	9	4	7
13	5	2	1
6	10	8	16
15	12	3	11

Определите минимальное число, N для которого это возможно. Представьте запись числа N в двоичной системе счисления, сложите те цифры этой записи, номер которых (нумеруя цифры с 0 от младшего разряда к старшему) делится нацело на 3 и запишите результат в десятичной системе счисления. Например, если $N=10101001_2$, ответом будет 2.

Примечания. Операция $A\%B$ вычисляет остаток от целочисленного деления A на B ; операция $A//B$ вычисляет частное от целочисленного деления A на B . При обращении к элементам матрицы первый индекс – номер строки, а второй – номер столбца, нумерация с (0,0).

Ответ: 12

Решение:

Заметим, что на каждом шаге цикла происходит обращение к значению $N\%16$, а потом деление $N//16$. Это не что иное как разбор на отдельные цифры шестнадцатеричной записи числа N . То есть, на каждом шаге цикла берется цифра с номером k (считая справа налево от 1) в записи числа в шестнадцатеричной системе счисления. Вспомним, что любая шестнадцатеричная цифра может быть представлена как пара цифр для записи в четверичной системе счисления, например $D_{16}=31_4$. Тогда $(N\%16)\%4$ – это младшая (вторая) цифра в такой паре, а $(N\%16)//4$ – старшая (первая) цифра в такой паре. Тогда операция $mas[(N\%16)\%4, (N\%16)//4]:=k$ означает, что в элемент матрицы, номер строки которого соответствует младшей цифре такой пары, а номер столбца – старшей цифры записывается номер соответствующей шестнадцатеричной цифры. Значит, мы можем брать последовательно элементы матрицы со значениями от 1 до 16, формировать для них пары (номер столбца, номер строки) и записывать такие пары справа налево (заметим, что нумерация идет от нуля):

32030001133312102230021120232131

Пользуясь тем, что любая четверичная цифра может быть представлена независимо как две двоичных, переведем в двоичную систему счисления и выделим разряды, номера которых (при нумерации с 0 от младшего к старшему разряду) делятся нацело на 3:

11100011000000101111110110010010000000101000101110011101

Посчитаем количество выделенных единиц – их будет ровно 12, что и является ответом на задание.

6. Телекоммуникационные технологии (2 балла)

[Три порта]

Петя моделирует работу прототипа коммутатора. У этого прототипа есть три порта. Два порта принимают данные из двух сетей со скоростью 5 КБайт в секунду каждый. Третий порт отдает данные в третью сеть со скоростью 10 КБайт в секунду. Данные из первой сети приходят пакетами по 40 КБайт, причем первый пакет начинает приниматься в начальный момент времени и далее новый пакет начинает приниматься по истечении каждых 10 секунд. Данные из второй сети приходят пакетами по 30 КБайт, причем первый пакет начинает приниматься по истечении 4 секунд от начального момента времени и далее новый пакет начинает приниматься по истечении каждых 8 секунд.

Прототип коммутатора работает следующим образом. Поступающий пакет по мере передачи данных записывается в буферную память. Если одновременно принимаются пакеты в оба порта их запись в буферную память происходит одновременно и независимо. Как только некоторый пакет целиком принят, он становится готов к передаче через выходной порт. Передача данных через выходной порт происходит исходя из полученных пакетов по 40 и по 30 КБайт соответственно, причем, если в некоторый момент времени есть возможность начать передавать пакеты обоих размеров, сначала передается пакет большего размера. В один момент времени через выходной порт может передаваться только один пакет и пакет всегда передается целиком. Передача пакета через выходной порт начинается сразу, как только появляется готовый пакет, если в этот момент не передается другой пакет или сразу после завершения передачи предыдущего пакета. Как только очередной пакет передан, он удаляется из буфера. Если в один момент времени нужно удалить переданный пакет и записать очередную порцию данных принимаемого пакета, сначала происходит удаление. Если вся память в буфере занята, но происходит попытка принять данные, происходит переполнение буфера и аварийная остановка коммутатора. Определите минимальный размер буфера в КБайтах, при котором в процессе работы данной модели никогда не произойдет аварийная остановка. В ответе укажите целое число.

Ответ: 90

Решение:

Вручную или с помощью электронной таблицы построим модель работы прототипа коммутатора. Одна строка означает состояние модели по истечении очередной секунды. В столбцах «Порт 1» и «Порт 2» отображается объем данных в КБайт, полученных через этот порт к очередному моменту времени. В столбце «Выход» показаны в виде отрицательных чисел данные, удаленные из буфера к очередному моменту времени. В столбце «Буфер» считается объем данных, хранимых в буфере в указанный момент. Желтые клетки означают секунду, на которой заканчивается приём очередного пакета. Со следующей секунды возможна его передача. Зеленые клетки означают секунды, на которых передается в третью сеть пакет, размером, 40 КБайт, а голубые – размером 30 КБайт.

Секунда	Порт 1	Порт 2	Выход	Буфер
1	5			5
2	10			10
3	15			15
4	20			20
5	25	5		30
6	30	10		40
7	35	15		50
8	40	20		60
9	40	25		65
10	40	30		70
11	45	30		75
12	50	30		80
13	55	35	-40	50
14	60	40	-40	60
15	65	45	-40	70
16	70	50	-70	50
17	75	55	-70	60
18	80	60	-70	70
19	80	60	-70	70
20	80	60	-70	70
21	85	65	-70	80

22	90	70	-70	90
23	95	75	-110	60
24	100	80	-110	70
25	105	85	-110	80
26	110	90	-140	60
27	115	90	-140	65
28	120	90	-140	70
29	120	95	-140	75

Обратим внимание, что по истечении 22-ой секунды от начального момента времени в буфере занято 90 КБайт. При этом, уже в начале следующей секунды из буфера будет удалено 40 КБайт, а поступит 10, причем по условию удаление будет предшествовать заполнению. Легко заметить, что достижение занятости в буфере 90 КБайт происходит в худшем случае, когда одновременно завершен на 18-й секунде прием сразу двух пакетов. В этом случае будет происходить накопление данных в буфере во время передачи большего пакета при наличии в памяти полного пакета меньшего размера. В принципе, можно продолжить таблицу и убедиться, что вторая такая же ситуация с одновременным завершением приёма двух пакетов произойдет по истечении 56-й секунды и, следовательно, занятость буфера в 90 КБайт следующий раз будет достигнута на 60-й секунде. Следовательно, процесс изменения памяти буфера будет цикличен и значение 90 КБайт никогда не будет превышена. Значит это и есть максимальный объем буфера, при котором будет неограниченная во времени безаварийная работа.

7. Технологии обработки информации в электронных таблицах, технологии сортировки и фильтрации данных (1 балл)

[Если...]

Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D	E
1		28	63	18	
2		=ЕСЛИ(ИЛИ(ОСТАТ(A2;B1)<>0;НЕ(И(ОСТАТ(A2;C1)=0;ОСТАТ(A2;D1)=0));ОСТАТ(A2;A1)=0);1;0)			
3					
4					

Какое максимальное целое число можно поместить в ячейку A1 так, чтобы при любом целом положительном числе в ячейке A2, в ячейке B2 выводилось значение 1. В ответе укажите целое число.

Ответ: 252

Решение:

Обозначим утверждение, что число в ячейке A2 делится нацело на некоторое целое число X как $DIV(X)$.

Тогда можно утверждать, что 1 в ячейке B2 будет выведено тогда и только тогда, когда будет истинным выражение:

$\overline{DIV(28)} \vee \overline{DIV(63)} \wedge \overline{DIV(18)} \vee DIV(A1)$, где A1 – значение в ячейке A1.

Преобразуем это выражение и получим:

$$\overline{DIV(28)} \vee \overline{DIV(63)} \vee \overline{DIV(18)} \vee DIV(A1)$$

Тогда, чтобы это выражение было истинно для любых значений в ячейке A2, утверждение $DIV(A1)$ должно быть эквивалентно утверждению $DIV(28) \wedge DIV(63) \wedge DIV(18)$. То есть A1 должно принимать такое значение, чтобы для тех чисел, которые одновременно делятся на 28, 63 и 18, они точно делились бы на A1. Найдем наименьшее общее кратное трех чисел: $28=2*2*7$, $63=3*3*7$, $18=2*3*3$. Получим $2*2*3*3*7=252$. Тогда, в качестве значения A1 нам подходит любой делитель числа 252, но в условии просят найти максимальное значение. Значит это само число 252.

8. Технологии программирования (2 балла)

[Рассадка в самолете]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

Вам дана схема самолета, со следующими обозначениями:

Символ «#» обозначает занятое место

«.» - свободное место

«|» - проход

«_» - перегородки между отсеками

Пример схемы самолета на шестнадцать мест (четыре свободных и двенадцать занятых) с одним проходом и двумя отсеками:

##|#.

```

.#|##
_|_
.#|##
####.

```

Координатой кресла будем считать номер строки и номер столбца, в которых это кресло содержится на схеме. Расстоянием между двумя креслами определим по формуле: $d=|x_1-x_0|+|y_1-y_0|+p+2w$, где (x_0, y_0) — координаты первого кресла, (x_1, y_1) — координаты второго кресла, p — число проходов между y_0 -м и y_1 -м столбцом, а w — число перегородок между x_0 -й и x_1 -й строкой.

Например, на схеме выше расстояние между левым верхним свободным креслом (2-я строка, 1-й столбец) и правым нижним креслом (5-я строка, 5-й столбец) будет равно $|5-2|+|5-1|+1+2\cdot1=10$.

Вам необходимо найти два свободных места с наибольшим расстоянием между ними и рассчитать это расстояние.

Формат входных данных

В первой строке входных данных задано число t — число схем самолетов ($1 \leq t \leq 100$).

Затем следует t описаний схем самолетов. В первой строке описания схемы содержатся два целых числа n и m — число строк и столбцов ($1 \leq n, m \leq 20$). В следующих n строках содержатся по m символов каждый из которых либо «#», либо «.», либо «_», либо «|».

Гарантируется, что есть хотя бы два свободных места, и если хотя бы один символ в строке «_», то все символы в строке задают перегородку либо проход. Также гарантируется, что если хотя бы один символ в столбце «|», то все символы в этом столбце задают проход.

Формат выходных данных

Для каждой схемы выведите расстояние между двумя искомыми местами в отдельной строке.

Пример

Стандартный ввод	Стандартный вывод
<pre> 4 5 5 ### . .# ## _ _ .# ## ### . 1 5 . . 5 5 .# ## _ _ _ _ ### . _ _ _ _ ### . 3 4 _ _ _ _ . . </pre>	<pre> 10 3 14 5 </pre>

Решение

Данная задача была представлена в двух вариантах. В первом нужно было найти два кресла с наибольшим расстоянием между ними, во втором с наименьшим.

Оба варианта решаются абсолютно аналогично. Более того, от участников не требовалось написать самое оптимальное решение. Ограничения позволяют перебрать два свободных кресла и линейным проходом найти расстояние между ними.

Например, для фиксированных свободных кресел (x_0, y_0) и (x_1, y_1) , можно было запустить два цикла:

- Первый от $\min(x_0, x_1)$ до $\max(x_0, x_1)$ внутри которого будем считать число встреченных символов «_» в столбце y_0 , например. Символов «|» в столбце y_0 быть не может, нам это гарантировано условием.

- Второй от $\min(y_0, y_1)$ до $\max(y_0, y_1)$, внутри которого будем считать число встреченных символов «|» в строке x_0 , например. Символов «_» в строке x_0 быть не может, нам это также гарантировано условием.

Зная число перегородок и проходов, можно вычислить расстояние подставив значения в формулу из условия. Далее остается всего лишь найти минимум/максимум среди всех посчитанных расстояний.

С примерами авторских решений можно ознакомиться в папке solutions в архиве задачи.

9. Технологии программирования (2 балла)

[Ближайшая интересная пара]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	3 секунды
Ограничение по памяти	256 мегабайт

Назовём пару элементов массива a_i и a_j интересной, если $i < j$, $a_i = -a_j$ и $a_i \leq 0$.

Вам дано несколько массивов, необходимо в каждом из них найти интересную пару с минимальной абсолютной разностью между индексами.

Формат входных данных

В первой строке дано число t — число тестовых наборов ($1 \leq t \leq 100$).

Затем следуют t тестовых наборов. Каждый тестовый набор задается двумя строками. В первой из них дано число n_i — количество элементов в i -м массиве ($1 \leq n_i \leq 10^5$). Во второй строке заданы n_i чисел $a_{i,j}$ — элементы i -го массива ($-10^9 \leq a_{i,j} \leq 10^9$).

Гарантируется, что сумма n_i по всем тестовым наборам не превосходит $5 \cdot 10^5$.

Формат выходных данных

Для каждого тестового набора выведите одно число — минимальную разность индексов в интересных парах. Если интересных пар в массиве нет, выведите -1 .

Пример

Стандартный ввод	Стандартный вывод
5	1
3	3
1 -1 1	1
4	-1
-1 2 -2 1	1
4	
1 -2 2 -1	
3	
1 2 3	
2	
0 0	

Решение

Данная задача была представлена в двух вариантах. В первом нужно было найти пары, где $i < j$, $a_i = -a_j$ и $a_i \leq 0$, а во $i < j$, $a_i = -a_j$ и $a_i \geq 0$. Оба варианта решаются абсолютно аналогично. Разберем первый вариант. Один из способов решения заключается в том, чтобы составить два массива b и c , таким образом, чтобы в b лежали пары (a_i, i) для $a_i \leq 0$, а в c пары (a_i, i) для $a_i \geq 0$. Затем можно отсортировать массив b по убыванию a_i , а c по возрастанию a_i . При этом порядок индексов для равных a_i должен быть возрастающим. Например, если $a = [-1, 2, -2, 1, 0, 0, -2]$, то b после сортировки будет $[(0, 4), (0, 5), (-1, 0), (-2, 2), (-2, 6)]$, а c — $[(0, 4), (0, 5), (1, 3), (2, 1)]$. Далее с помощью метода двух указателей можно пройти по этим двум массивам и найти нужную пару. Подробнее с примером такого прохода в авторском решении можно ознакомиться в папке solutions в архиве задачи.

Отборочный этап. Первый тур (приведен один из вариантов заданий)

1. Кодирование информации. Системы счисления (2 балла)

[Степени]

Записан ряд чисел в шестнадцатеричной системе счисления. Каждое число, это 21_{16}^N , где N принимает значения всех натуральных чисел от 1 до X. Все числа попарно различны. Для всех чисел просуммировали два их младших разряда и получили общую сумму двух младших разрядов всех записанных чисел. Записав это значение в десятичной системе счисления, получили число 392. Определите и запишите в ответ значение X.

Ответ: 50

2. Кодирование информации. Системы счисления (1 балл)

[Опять степени]

Дано равенство:

$$20_{16}^N - 10_{16}^N - 4_{16}^N = X_2$$

где, N – целое положительное число, а X – результат вычисления левой части равенства, записанный в двоичной системе счисления. Известно, что X содержит одну последовательность из ровно 20 идущих подряд единиц так, что слева и справа от неё находятся нули или границы записи числа. X может также содержать последовательности из большего или меньшего количества единиц. Найдите и запишите в ответ через пробел в порядке возрастания все значения N, при которых это возможно.

Ответ: 10 21

3. Кодирование информации. Количество информации. Кодирование текста (1 балл)

[Пароли, пароли]

Петя написал генератор паролей, длиной 8 символов. Каждый символ с равной вероятностью может быть заглавной или строчной латинской буквой, арабской цифрой или одним из двух специальных значков: # и @. Сколько бит информации несет в себе сообщение, что сгенерированный пароль не содержит ни одной буквы и цифры? В ответе укажите целое число.

Ответ: 40

4. Кодирование информации. Объем данных (1 балл)

[Цифровая фоторамка]

Петя разрабатывает прототип цифровой фоторамки, демонстрирующей слайдшоу. Петя пока не умеет сжимать данные, поэтому каждый слайд он хранит в памяти фоторамки как последовательность кодов пикселей, используя для каждого кода минимально возможное одинаковое для всех пикселей количество бит. Петя не сохраняет никакой дополнительной информации о слайдах, поскольку использует только размер 1024 на 768 пикселей и палитру из 65536 цветов. Закончив с модулем воспроизведения слайдов, Петя решил дописать модуль для проигрывания музыкального сопровождения. Для этого в памяти фоторамки сохраняется без сжатия одноканальная музыкальная композиция с частотой дискретизации 22 кГц и 65536 уровнями квантования. Продолжительность композиции равна продолжительности однократного показа всей последовательности сохраненных слайдов при учете, что каждый слайд демонстрируется ровно 8 секунд. Программа Пети заняла в памяти фоторамки 256 КБайт. Определите, максимальное количество слайдов в слайдшоу со звуковым сопровождением, если известно, что память фоторамки составляет 30 МБайт.

Примечание: 1 КБайт = 1024 байта, 1 МБайт = 1024 КБайта.

Ответ: 16

5. Основы логики. Анализ логических функций (2 балла)

[Три xor]

Вариант 1

Про логическую функцию от трех переменных A, B и C известно следующее:

1. Если $(A \text{ xor } B)$ примет значение «истина», то функция примет такое же значение как $(\text{not } B \text{ or not } C)$.
2. Если $(A \text{ xor } C)$ примет значение «истина», то функция примет такое же значение как $(C \rightarrow A \text{ and } B)$.
3. Если $(B \text{ xor } C)$ примет значение «ложь», то функция примет такое же значение как $(A \text{ or not } B)$.

Найдите эту функцию и запишите в ответ максимально упрощенную формулу для этой функции, содержащую только операции инверсии, конъюнкции и дизъюнкции. Запись может содержать скобки. Будем считать формулу максимально упрощенной, если не существует эквивалентной ей формулы, содержащей меньшее количество логических операций.

Комментарий по вводу ответа: операнды вводятся большими латинскими буквами; логические операции обозначаются, соответственно как **not**, **and** и **or**.

Скобки используются только для изменения порядка выполнения операций. Если порядок выполнения операций очевиден из их приоритетов – дополнительное использование скобок считается ошибкой.

Пример записи ответа: $(A \text{ or not } B) \text{ and } C$

Ответ: $A \text{ or not } C \parallel \text{not } C \text{ or } A$

6. Основы логики. Упрощение логического выражения (2 балла)

[Эквиваленции]

Упростите логическое выражение или укажите его результат (при его однозначности). Результат упрощения может содержать только операции инверсии, конъюнкции и дизъюнкции и не должен содержать скобок.

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

Комментарий по вводу ответа: операнды вводятся большими латинскими буквами; логические операции обозначаются, соответственно как **not**, **and** и **or**.

При однозначном ответе – истинный ответ обозначается как 1, а ложный как 0.

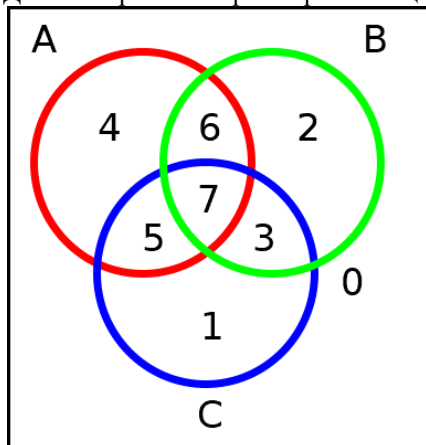
Пример записи ответа: $A \text{ or } B \text{ and not } C$

Ответ: **not A or not B or not C or D**

7. Основы логики. Синтез выражения по логической схеме (2 балла)

[Цветные окружности]

Дано изображение трех пересекающихся окружностей:



На изображении выделены восемь непересекающихся пронумерованных областей (область, пронумерованная нулем, означает точки, не принадлежащие ни одной из окружностей).

Для этого изображения сформулированы три логических высказывания:

$A = \{\text{точка находится внутри красной окружности}\}$

$B = \{\text{точка находится внутри зеленой окружности}\}$

$C = \{\text{точка находится внутри синей окружности}\}$

Известно, что некоторое логическое выражение, содержащее одно или несколько перечисленных логических высказываний и одну или несколько логических операций из набора {инверсия, конъюнкция, дизъюнкция}, принимает истинное значение только для точек, находящихся в областях с номерами 2, 4, 5, 6, 7. Определите это логическое выражение и запишите в ответ максимально упрощенную формулу для него. Запись может содержать скобки. Будем считать формулу максимально упрощенной, если не существует эквивалентной ей формулы, содержащей меньшее количество логических операций.

Комментарий по вводу ответа: операнды вводятся большими латинскими буквами; логические операции обозначаются, соответственно как **not**, **and** и **or**.

Скобки используются только для изменения порядка выполнения операций. Если порядок выполнения операций очевиден из их приоритетов – дополнительное использование скобок считается ошибкой.

Пример записи ответа: $(A \text{ or not } B) \text{ and } C$

Ответ: **$A \text{ or } B \text{ and not } C \parallel A \text{ or not } C \text{ and } B \parallel B \text{ and not } C \text{ or } A \parallel \text{not } C \text{ and } B \text{ or } A$**

8. Алгоритмизация и программирование. Формальный исполнитель (2 балла)

[Волшебные грядки]

Урфин Джюс обрабатывает грядки с волшебными растениями. Каждое утро Урфин Джюс просыпается и если видит хотя бы одну необработанную грядку, то обрабатывает за день ровно одну грядку. Проснувшись в первый день, Урфин Джюс увидел 404 необработанные грядки. После того, как Урфин Джюс заканчивает обрабатывать очередную грядку, он идет спать, но пока он спит, могут произойти следующие события:

1. После обработки Урфином Джюсом каждой десятой грядки прилетает злая волшебница Бастинда и добавляет 15 необработанных грядок.

2. После обработки Урфином Джюсом каждой третьей грядки прилетает добрая волшебница Виллина и обрабатывает 10 необработанных грядок (или все оставшиеся, если осталось меньше десяти необработанных грядок).

Если эти два события происходят одновременно, то добро побеждает зло, поэтому Виллина прогоняет Бастинду и обрабатывает как обычно 10 необработанных грядок (или все оставшиеся, если осталось меньше десяти необработанных грядок), а для Бастинды отсчет 10 грядок начинается заново.

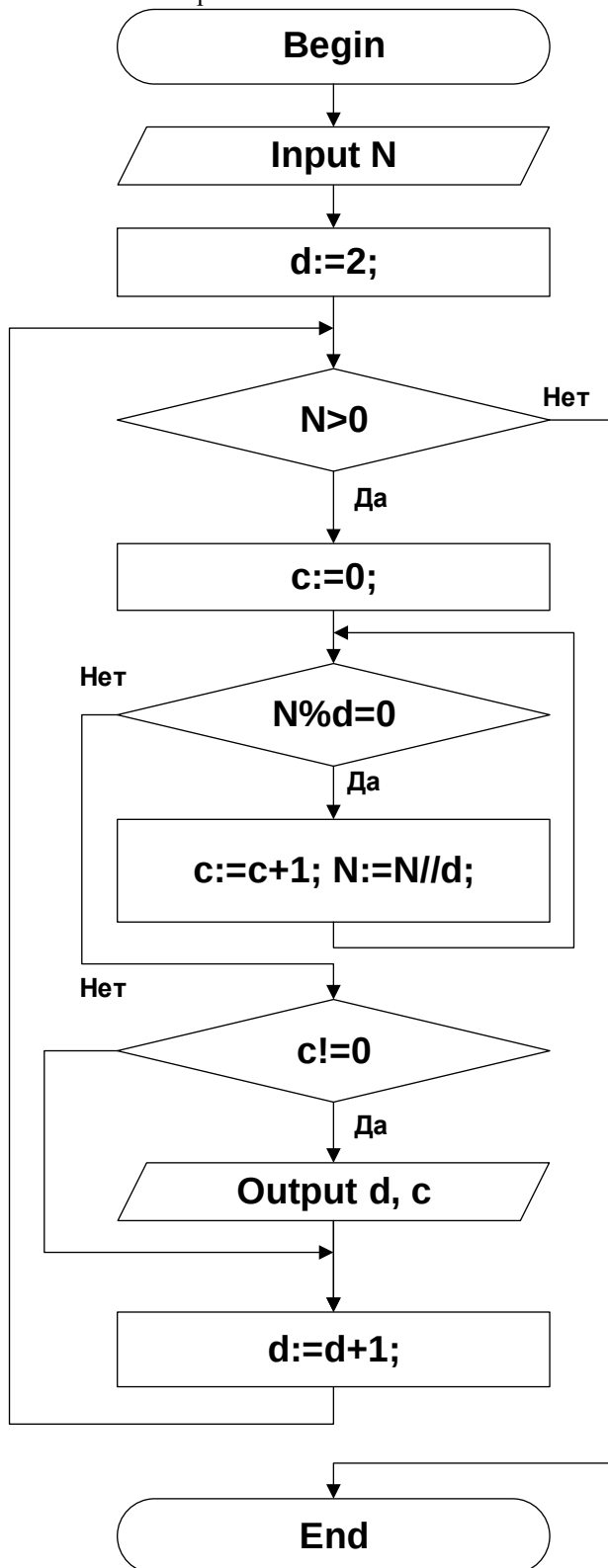
Напишите номер дня, утром которого Урфин Джюс проснется и не увидит ни одной необработанной грядки.

Ответ: **124**

9. Алгоритмизация и программирование. Блок-схема, обратная задача (1 балл)

[Пары чисел]

Дана блок-схема алгоритма:



На вход подали натуральное число N. Известно, что в результате выполнения алгоритма было выведено:

2 3
5 2
7 4

Найдите и укажите в ответе значение N, при котором это могло произойти. Если такого значения не существует, укажите в ответе NULL.

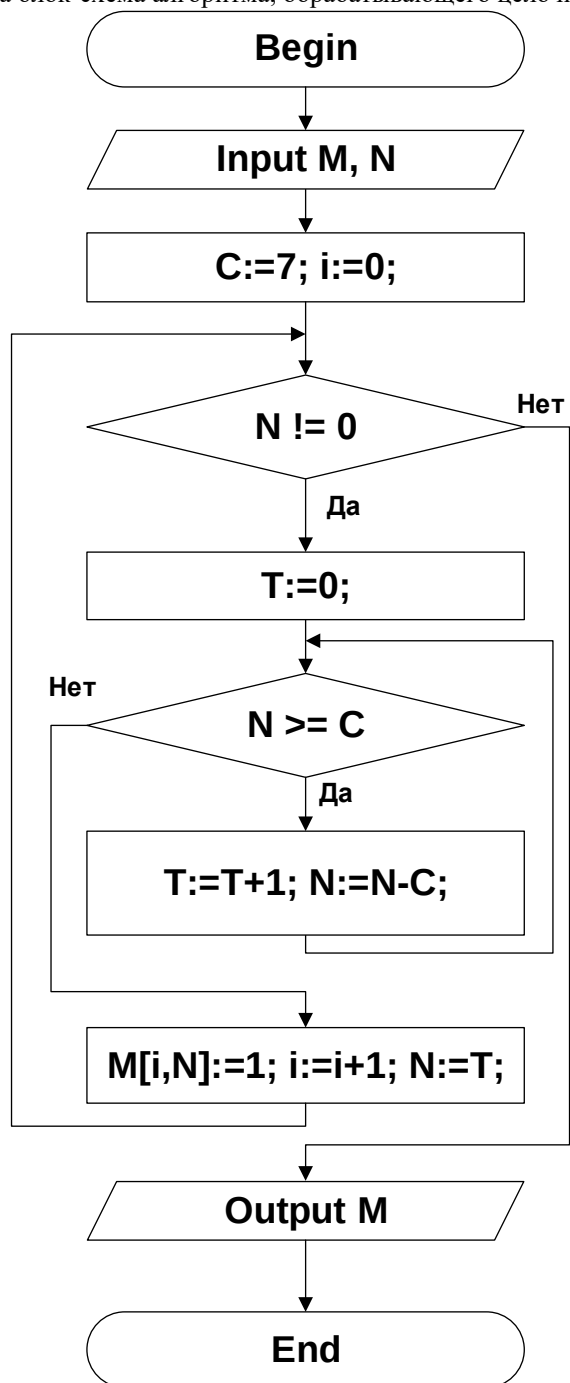
Примечание: Оператор «!=» означает «не равно». Операция A%B означает получение остатка от целочисленного деления A на B. Операция A//B означает получение частного от целочисленного деления A на B.

Ответ: 480200

10. Алгоритмизация и программирование. Блок-схема, массивы (2 балла)

[Пороги]

Дана блок-схема алгоритма, обрабатывающего целочисленную матрицу, размером 7 на 7 элементов:



На вход подали целое положительное число N и матрицу M, заполненную только нулевыми значениями:

На выходе получили следующую матрицу M:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

При каком значении N это возможно? В ответе укажите целое положительное число. Если такого числа не существует, в ответе напишите NULL.

Примечания:

1. При обращении к элементам матрицы первый индекс означает номер строки, а второй – номер столбца. Нумерация производится от 0.
2. Оператор «!=» означает «не равно».

Ответ: 123446

Отборочный этап. Второй тур (приведен один из вариантов заданий)

1. Электронные таблицы. Адресация ячеек и вычисления (2 балла)

[Шашечки]

Дан фрагмент электронной таблицы в режиме отображений формул:

	A	B	C
1		=ЕСЛИ(A1>2;A1-3;0)	
2	=ЕСЛИ(A1>4;A1-5;0)	=ОСТАТ(\$A2*B\$1;2)	
3			
4			

Ячейку B1 скопировали во все ячейки диапазона C1:CW1. Ячейку A2 скопировали во все ячейки диапазона A3:A61. Ячейку B2 скопировали во все ячейки диапазона B2:CW61. В ячейку A1 поместили целое положительное число. Известно, что сумма значений всех ячеек диапазона B2:CW61 равна 814. Определите, сколько существует значений ячейки A1, при которых это возможно и максимальное из них. В ответе укажите через пробел два числа: сначала количество возможных значений, а затем максимальное возможное значение.

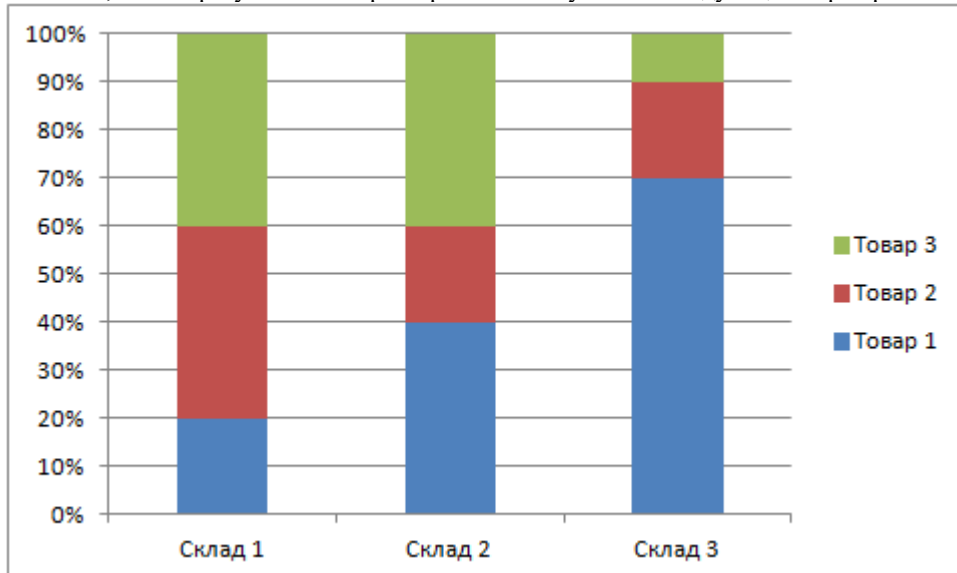
Ответ: 6 227

2. Электронные таблицы. Графики и диаграммы (1 балл)

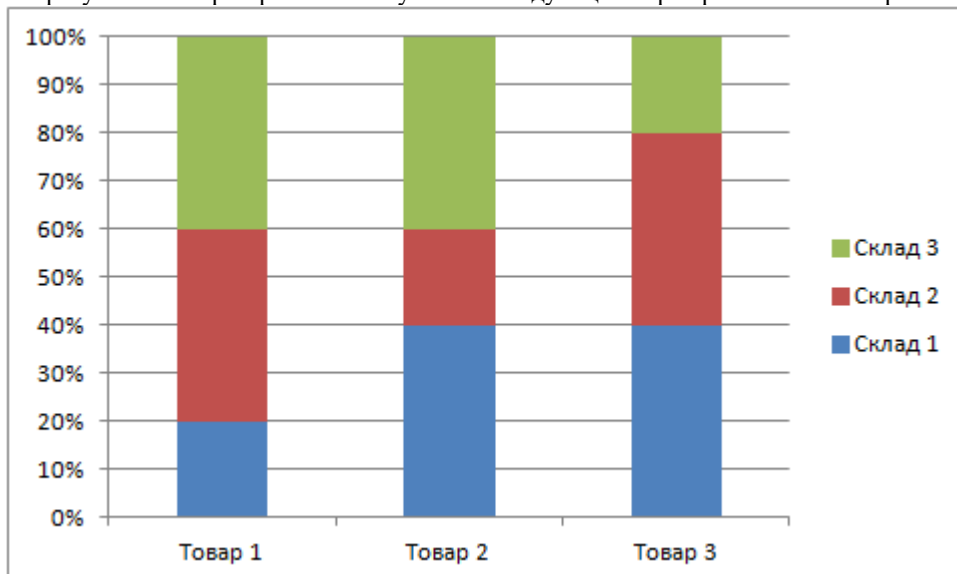
[Кража со взломом]

В компании 3 склада. На каждом складе хранится целое количество каждого из трех товаров. Во время инвентаризации по данным о количестве каждого товара на каждом складе строится нормированная гистограмма с накоплением. Поскольку при построении такой диаграммы можно менять местами оси на диаграмме, каждый ревизор использует один из двух вариантов её построения.

Известно, что по результатам первой ревизии получилась следующая нормированная гистограмма с накоплением:



А в результате второй ревизии получилась следующая нормированная гистограмма с накоплением:



Известно, что между ревизиями с одного склада было украдено некоторое количество единиц только одного товара. Также известно, что количество товаров на остальных складах не изменилось. Еще известно, что на Складе 1 хранится 40 единиц Товара 1. Определите, с какого склада украден какой товар и в каком количестве. В ответе укажите через пробелы три числа: сначала номер склада, затем номер товара, а затем количество украденных единиц товара.
Ответ: 3 1 200

3. Сортировка и фильтрация данных (1 балл) [Двоечники]

В базе данных хранятся сведения об успеваемости студентов по определенной дисциплине. Известно, что всего на курсе учится 287 студентов, которых разделили на 3 потока, пронумерованных от 1 до 3. Среди прочего, для каждого из этих студентов хранятся сведения о номере его потока и набранных по этой дисциплине баллах (целое число от 0 до 100 включительно). Известно количество записей, полученных в результате выполнения следующих запросов:

Запрос	Количество записей
Выбрать Табельный_номер_студента где Номер_потока больше 1 и Балл больше 60	129
Выбрать Табельный_номер_студента где Номер_потока меньше 3 и Балл меньше 75	63
Выбрать Табельный_номер_студента где Номер_потока равно 1 и Балл больше 74	69

Какое минимальное количество записей может получиться в ответ на запрос:

«Выбрать Табельный_номер_студента где Номер_потока равно 3 и Балл меньше 61»?

В ответе укажите целое число.

Ответ: 26

4. Мультимедиа технологии, сортировка данных (2 балла) [Цветосортировка]

Исходное изображение состоит из 12 последовательных квадратов, каждый из которых залит некоторым цветом:



Все цвета закодированы с помощью цветовой модели RGB.

Для сортировки этих квадратов использовали следующий алгоритм:

1. Сначала все квадраты отсортировали по возрастанию значению одной из цветовых координат.
2. Затем квадраты, у которых были одинаковыми значения цветовой координаты, по которой отсортировали в предыдущем пункте, между собой отсортировали по убыванию другой цветовой координаты.
3. Наконец, квадраты, у которых оказались одинаковыми значения и цветовой координаты, по которой сортировали в первом пункте и цветовой координаты, по которой сортировали во втором пункте, между собой отсортировали по возрастанию оставшейся цветовой координаты.

Алгоритм применяли к исходному изображению независимо два раза и получили два изображения с отсортированными квадратами:



Определите для каждого из этих изображений порядок использования цветовых координат в алгоритме сортировки.

Например, если для получения некоторого изображения в первом пункте алгоритма использовалась цветовая координата R, во втором пункте – цветовая координата B, а в третьем пункте – цветовая координата G, то такой порядок может быть обозначен как RBG.

Запишите в ответ через пробел сначала обозначение для порядка сортировки, с помощью которого было получено первое изображение, а затем обозначение для порядка сортировки, с помощью которого было получено второе изображение.

Ответ: GBR RGB

5. Телекоммуникационные технологии (3 балла). [Три помехи]

Через сетевое соединение необходимо передать 384 КБайт данных от источника к приемнику. Передача данных осуществляется следующим образом. Весь объем данных разбивается на блоки по N КБайт. Поскольку N – целое число, последний блок может получиться меньше, чем N КБайт. К каждому блоку данных добавляется служебная информация. Она фиксированная и составляет 1 КБайт. В результате получается пакет, который передается по сети со скоростью 1 КБайт в секунду. Как только пакет получен, он проходит проверку на целостность, по результатам которой источнику отправляется ответ. Проверка пакета и доставка ответа занимают ровно 1 с. Если приходит ответ, что пакет передан корректно, то немедленно начинается передача следующего пакета. Если приходит ответ, что пакет передан некорректно, то немедленно начинается повторная передача этого же пакета. Будем считать, что ответ всегда передается корректно (в силу небольшого размера и использования помехоустойчивого кодирования), даже если во время передачи ответа была помеха.

Передача считается завершенной тогда, когда источник получает ответ от приемника, что последний переданный пакет передан корректно.

В процессе передачи было три помехи, которые приводили к тому, что, если в момент помехи шла передача пакета, после завершения его передачи, обнаруживалось, что он передан некорректно. Помеха носит мгновенный характер, но гарантированно приводит к некорректности переданного пакета, если попадает на время его передачи.

Известно, что первая помеха произошла через 64,5 секунды от начала передачи данных, вторая помеха произошла через 116,5 секунд от начала передачи данных, а третья помеха произошла через 337,5 секунд от начала передачи данных.

Определите значение N , при котором передача данных завершится за минимально возможное для передачи указанного объема данных при указанном количестве помех время и рассчитайте это время. Если таких значений N несколько, выберите минимальное из них. В ответе укажите через пробел два целых числа: сначала значение N , а затем вычисленное минимальное время передачи данных.

Ответ: 11 454

6. Операционные системы (2 балла)

[Каскад]

Петя экспериментирует с параллельным исполнением процессов в операционной системе. Он написал скрипт, реализующий следующую модель.

В начальный момент времени и далее по истечении каждых 2-х секунд рождается новый процесс. При рождении процессу выделяется одна страница памяти. Далее, по истечении каждой секунды каждому процессу мгновенно удваивается выделенное количество страниц памяти. Если истекло N секунд от рождения процесса, то вместо удвоения объема памяти процесс завершается и освобождает всю память, которую он занимал. Операционная система всегда сначала завершает процессы, для которых наступило время завершения и освобождает занимаемую ими память, а уже потом выделяет память другим процессам, если эти операции приходится на один и тот же момент времени. Если в какой-то момент времени обнаружится, что свободной в этот момент памяти не хватит на удвоение количества страниц памяти всех активных процессов, происходит аварийная остановка работы скрипта. Петя знает, что в начальный момент времени у него свободно 22000 страниц памяти и никакие другие процессы, кроме порождаемых его скриптом не могут расходовать эту память. При каком максимальном значении N скрипт Пети может работать неограниченно долго? В ответе укажите целое число.

Ответ: 15

7. Технологии программирования (2 балла)

[К ближайших точек]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

Дано n точек на числовой прямой. Про каждую точку известна ее координата x_i .

Расстоянием между точками с координатами x_i и x_j будем считать абсолютную разность их координат: $|x_i - x_j|$. Точка с координатой x_j находится ближе к точке x_i , чем точка x_k , если расстояние между точками x_i и x_j меньше, чем между x_i и x_k .

Требуется выбрать из данных точек точку с наименьшей координатой такую, что сумма ее расстояний до ближайших к ней k точек наименьшая возможная.

Формат входных данных

В первой строке дано два натуральных числа n и k — число точек и число ближайших точек, которые надо учитывать ($1 \leq k < n \leq 10^5$).

Во второй строке заданы n целых чисел x_i — координаты точек ($-10^9 \leq x_i \leq 10^9$).

Формат выходных данных

Выведите два числа — координату выбранной точки и сумму расстояний от нее до ближайших к ней k точек.

Примеры

Стандартный ввод	Стандартный вывод
5 2 4 2 3 6 0	3 2
2 1 1 2	1 1
3 1 1 1 1	1 0

8. Технологии программирования (2 балла)

[Ближайшая интересная пара]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

Илья работает баскетбольным аналитиком. Сегодня после игры, вместо традиционного протокола, где для каждого игрока известна полная статистика, организаторы игры выдали Илье протокол в формате «Ход игры».

В протоколе для каждого броска в игре записаны фамилия игрока, совершившего бросок, сколько очков он мог набрать в случае попадания, а также результат попытки: забил или нет.

Броски бывают трех номиналов:

Двухочковый бросок - за попадание начисляется два очка, в протоколе обозначается «2pt»

Трехочковый бросок - за попадание начисляется три очка, в протоколе обозначается «3pt»

Штрафной бросок - за попадание начисляется одно очко, в протоколе обозначается «FT» (сокращенно от Free Throw)

Отметка «Made» означает, что бросок был забит, а отметка «Missed», что бросок был промазан. Таким образом запись в протоколе о том, что игрок с фамилией James забил двухочковый бросок будет выглядеть так: «James 2pt Shot: Made». Если же игрок с фамилией Westbrook промажет трехочковый бросок, то запись будет «Westbrook 3pt Shot: Missed».

Полной статистикой игрока назовем данные о том, сколько бросков каждого номинала он забил из скольких попыток и сколько при этом очков он набрал. Например, «1/2 2pt, 1/3 3pt, 0/0 FT, Total: 5 Points»

Илья хочет выделить игрока, который забил больше всех бросков. Помогите Илье найти такого игрока и посчитать его полную статистику. Если таких игроков несколько, выведите всех, упорядоченных лексикографически по фамилиям.

Формат входных данных

В первой строке дано число n - число записей в протоколе ($1 \leq n \leq 1000$).

В следующих n строках содержатся записи протокола в формате, описанном выше. Гарантируется, что у всех игроков уникальные фамилии, состоящие из букв латинского алфавита. Гарантируется, что длина каждой фамилии не превосходит 20 символов.

Порядок исполнения штрафных бросков может противоречить реальным правилам баскетбола.

Формат выходных данных

В первой строке выведите число игроков, которые забili больше всех бросков. Далее для каждого игрока в отдельной строке выведите его фамилию и полную статистику разделенные пробелом. При этом фамилии должны быть упорядочены лексикографически.

Примеры

Стандартный ввод	Стандартный вывод
8 James 2pt Shot: Made Westbrook 3pt Shot: Made James 2pt Shot: Made Westbrook 3pt Shot: Missed Westbrook 3pt Shot: Missed Westbrook 3pt Shot: Missed Westbrook 3pt Shot: Missed Westbrook 3pt Shot: Made	2 James 2/2 2pt, 0/0 3pt, 0/0 FT, Total: 4 Points Westbrook 0/0 2pt, 2/6 3pt, 0/0 FT, Total: 6 Points
8 Ivanov FT Shot: Made Petrov 2pt Shot: Missed Ivanov FT Shot: Made Petrov 3pt Shot: Made Kuznetsov 3pt Shot: Missed Petrov 2pt Shot: Missed Ivanov 3pt Shot: Missed Sidorov 2pt Shot: Made	1 Ivanov 0/0 2pt, 0/1 3pt, 2/2 FT, Total: 2 Points

Задания для 5–8 класса

Заключительный этап (приведен один из вариантов заданий)

1. Основы логики, моделирование (1 балл)

[Настольные игры]

Как-то четверо мальчиков, живущих на одной улице, договорились собраться вместе и поиграть в настольные игры. Каждый мальчик обещал принести свою настольную игру.

Мальчиков зовут: Петя, Ваня, Тимур и Антон.

Настольные игры у них такие: Уно, Манчкин, Каркасон и Имаджинариум, у каждого мальчика ровно одна игра из перечисленных.

Дома, в которых живут мальчики пронумерованы от 1 до 4, все мальчики живут в разных домах.

Известно:

1. У мальчика из второго дома Уно
2. У Тимура Каркасон
3. Ни Петя, ни Ваня не живут во втором доме
4. У мальчика из четвертого дома Имаджинариум
5. У Вани не Имаджинариум

Так же мальчики не смогли договориться, в чьём доме им лучше собраться, поэтому решили выбрать дом, до которого всем будет ближе, однако знают лишь расстояние между домами. Ближайшим для всех домов является тот дом, для которого максимальное расстояние до других домов является минимальным среди остальных домов.

Вам дана информация о расстоянии между домами, записанная в виде таблицы:

	1	2	3	4
1	0	7	8	6
2	7	0	6	5
3	8	6	0	4
4	6	5	4	0

В такой таблице на пересечении некоторой строки и столбца записывается расстояние между двумя домами.

Помогите мальчикам определиться, где же им играть в настольные игры. В ответе напишите первую букву имени мальчика(заглавную), в чьём доме лучше всего проводить игры. Если существует более одного дома, который подпадает под определение ближайшего, в ответе можно указать первую букву имени мальчика, живущего в любом из таких домов

Ответ: П

Решение:

Для начала найдём номер ближайшего дома, записав максимальное число в каждой строке

	1	2	3	4	max
1	0	7	8	6	8
2	7	0	6	5	7
3	8	6	0	4	8
4	6	5	4	0	6

Таким образом дом, для которого наибольшее расстояние до другого дома, является наименьшим среди остальных – дом с номером 4. Осталось только узнать, кто из мальчиков живет в этом доме.

Для этого построим таблицу:

	1	2	3	4	У	М	К	И
П								
В								
Т								
А								
У								
М								
К								
И								

В такой таблице красным отмечены номера домов, желтым – первые буквы имен мальчиков, зеленым – первые буквы названий игр. Будет расставлять + на пересечении соответствующих строк и столбцов, если есть связь. В противном случае ставим -. Согласно условию, все мальчики живут в различных домах и у каждого своя игра.

По пунктам будем заполнять таблицу.

1. У мальчика из второго дома Уно

	1	2	3	4	У	М	К	И
П								
В								
Т								
А								
У	-	+	-	-				
М		-						
К		-						
И		-						

2. У Тимура Каркасон, следовательно, Тимур не из второго дома.

	1	2	3	4	У	М	К	И
П							-	
В							-	
Т		-			-	-	+	-
А							-	
У	-	+	-	-				
М		-						
К		-						
И		-						

3. Ни Петя, ни Ваня не живут во втором доме. Следовательно, Антон живет во втором доме. И у Антона Уно.

	1	2	3	4	У	М	К	И
П		-			-		-	
В		-			-		-	
Т		-			-	-	+	-
А	-	+	-	-	+	-	-	-
У	-	+	-	-				
М		-						
К		-						
И		-						

4. У мальчика из четвертого дома Имаджинариум. Следовательно, Тимур не живет в 4 доме.

	1	2	3	4	У	М	К	И
П		-			-		-	
В		-			-		-	
Т		-		-	-	-	+	-
А	-	+	-	-	+	-	-	-
У	-	+	-	-				
М		-		-				
К		-		-				
И	-	-	-	+				

5. У Вани не Имаджинариум. Значит, Имаджинариум у Пети. Тогда у Вани Манчкин. А Петя живет в четвертом доме

	1	2	3	4	У	М	К	И
П	-	-	-	+	-	-	-	+
В		-		-	-	+	-	-
Т		-		-	-	-	+	-
А	-	+	-	-	+	-	-	-
У	-	+	-	-				
М		-		-				
К		-		-				
И	-	-	-	+				

Поскольку Петя живет в четвертом доме, то ответ на поставленную задачу первая буква имени Пети.

2. Моделирование, графы (1 балл)

[Вася снова в лабиринте.]

Вася выбрался из предыдущего подземелья, но через некоторое время он снова захотел попробовать свои силы в решение головоломки с лампочками, поэтому отправился в то подземелье вновь, но вот незадача, подземелье изменилось, однако правила остались теми же.

Подземелье состоит из комнат и туннелей между ними. В каждой комнате находится по 4 лампочки. В подземелье есть два типа комнат – красные и зеленые. В зеленых комнатах есть кнопка для включения лампочек, а в красных – нет.

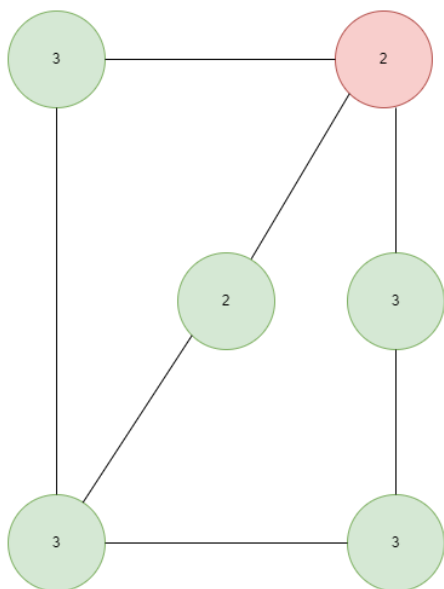
С помощью нажатия на кнопку можно включать лампочки по следующему правилу: одно нажатие на кнопку включает по одной лампочке в текущей комнате и во всех соседних комнатах, если в какой-то комнате уже горят 4 лампочки, то при попытке включить еще одну лампочку в этой комнате потухнут 3 лампочки и останется гореть одна.

Вася может нажимать на кнопки в любом порядке, а также беспрепятственно передвигаться по подземелью и заходить в любую комнату, в том числе начинать с любой комнаты.

Вам дан план подземелья. Комнаты изображены на плане как круги, а число внутри круга – количество горящих лампочек на момент попадания Васи в подземелье. Вася понял, что не может самостоятельно выбраться, поэтому помогите Васе узнать минимальное количество нажатий на кнопки, чтобы в каждой комнате горело ровно по 4 лампочки. Также мальчик пообещал сам себе что, если ему удастся выбраться, он больше не пойдет в это подземелье.

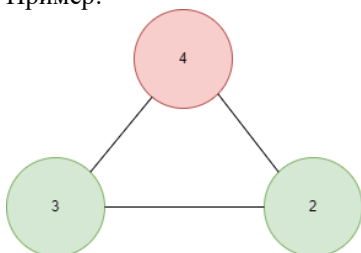
В ответе укажите одно число – минимальное количество нажатий кнопок, чтобы в каждой комнате было зажжено 4 лампочки.

Карта подземелья



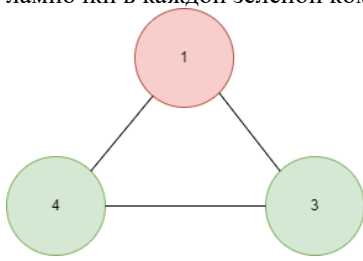
Соседней комнатой называется любая комната, которая соединена туннелем с текущей комнатой.

Пример:



Вася может нажимать кнопки только в зеленых комнатах.

Если Вася нажмет на кнопку в комнате с 3 лампочками, то потухнет 3 лампочки в красной комнате, и загорится по одной лампочке в каждой зеленой комнате.



Ответ: 8

Решение:

Давайте обозначим буквами латинского алфавита количество нажатий на кнопки в комнатах, нумеруя их слева направо и сверху вниз. Так за **a** возьмем количество нажатий на кнопку в левой верхней комнате, за **b** количество нажатий на кнопку в правой верхней комнате и так далее.

Таким образом можно составить систему уравнений, определяющую количество зажжённых лампочек, зная начальное количество. Чтобы горели все лампочки, нужно чтобы сумма была кратка четырем:

$$\begin{cases} (3 + a + b + e) \bmod 4 = 0 \\ (2 + a + b + c + d) \bmod 4 = 0 \\ (2 + b + c + e) \bmod 4 = 0 \\ (3 + b + d + f) \bmod 4 = 0 \\ (3 + a + c + e + f) \bmod 4 = 0 \\ (3 + d + e + f) \bmod 4 = 0 \end{cases}$$

Из условия известно, что **b=0**, так как комната красная и в ней нет кнопки, чтобы её нажать. Так же можно заметить из данной системы уравнений, что не имеет смысла нажимать на одну и ту же кнопку больше 3 раз.

Давайте избавимся от операции взятия остатка от деления. Для этого можно приравнять выражения в скобках к возможным значениям (справа от равно просто запишем возможные значения суммы через косую черту, учитывая ограничение на переменные от 0 до 3).

$$\begin{cases} 3 + a + e = 4/8 \\ 2 + a + c + d = 4/8 \\ 2 + c + e = 4/8 \\ 3 + d + f = 4/8 \\ 3 + a + c + e + f = 4/8/12 \\ 3 + d + e + f = 4/8 \\ b = 0 \end{cases}$$

Оставим в левой части равенства только переменные, в правой только возможные суммы.

$$\begin{cases} a + e = 1/5 \\ a + c + d = 2/6 \\ c + e = 2/6 \\ d + f = 1/5 \\ a + c + e + f = 1/5/9 \\ d + e + f = 1/5 \\ b = 0 \end{cases}$$

Начнём решать систему уравнений:

Из 4 и 6 уравнения следует, что $e=0/4$, но поскольку не имеет смысла количество нажатий больше 3, то **e=0**. Преобразуем систему.

$$\begin{cases} a = 1 \\ a + c + d = 2/6 \\ c = 2 \\ d + f = 1/5 \\ a + c + f = 1/5/9 \\ e = 0 \\ b = 0 \end{cases}$$

Получаем уже 4 известных переменных, подставим их в систему:

$$\begin{cases} a = 1 \\ 3 + d = 2/6 \\ c = 2 \\ d + f = 1/5 \\ 3 + f = 1/5/9 \\ e = 0 \\ b = 0 \end{cases}$$

Если в правых частях уравнений 2 и 5 оставить только минимальные возможные значения, то получаем $d=3$, $f=2$. Таким образом наше минимальное решение:

$$\begin{cases} a = 1 \\ b = 0 \\ c = 2 \\ d = 3 \\ f = 2 \\ e = 0 \end{cases}$$

Поскольку спрашивалось суммарное количество нажатий, то ответ получаем как $1+2+3+2=8$.

3. Кодирование информации, комбинаторика, регулярные выражения (3 балла) [Где взять столько сотрудников?]

Для задания регулярных выражений приняты следующие обозначения:

C	Любой неспециальный символ c соответствует самому себе. Специальными символами будем считать только символы [,], {, }, *, +, -, ? – эти символы не могут по условию данной задачи встретиться в тексте.
[...]	Любой символ из ...; допустимы диапазоны типа: • a-z (последовательно идущие символы в алфавите), • A-Z (последовательно идущие символы в алфавите), • 0-9 (последовательно идущие цифры). Диапазоны могут быть указаны друг за другом.
r*	Ноль или более вхождений символа r , может применяться и для диапазонов. Например, a* означает ноль или более вхождений символа a .
r+	Одно или более вхождений символа r , может применяться и для диапазонов, например [a-z]+ означает одно или более вхождений символов диапазона a-z в любом порядке.
r1r2	За символом или диапазоном r1 следует символ или диапазон r2 .
{n}	Число вхождений n предыдущего выражения. Например, выражение [a-z]{3} соответствует подстроке из трех латинских букв.
{n, m}	Число вхождений от n до m предыдущего выражения. Например, выражение [a-z]{3, 5} соответствует подстроке из не менее трех и не более пяти латинских букв.

В банке все e-mail адреса сотрудников задаются с помощью следующего регулярного выражения
[A-Z].[a-z]{1, 8}@sun.ru

Для каждого отдела банка имеется свой формат e-mail адресов, который не противоречит общепринятому регулярному выражению для e-mail. У Кирилла свой отдел, сколько сотрудников может быть в его отделе, если у каждого его сотрудника должен быть e-mail, подходящий под регулярное выражение для отдела Кирилла **[A-E].[a-z]+kir@sun.ru**

В ответе укажите одно число – максимальное количество сотрудников в отделе Кирилла.

Ответ: 61783150

Решение:

Согласно условию, все адреса сотрудников банка задаются с помощью регулярного выражения **[A-Z].[a-z]{1, 8}@sun.ru**, в отделе Кирилла же адреса e-mail задаются с помощью регулярного выражения **[A-E].[a-z]+kir@sun.ru**. Чтобы получить регулярное выражение, попадающее под оба вышеуказанных регулярных выражения, нужно разделить их на части и брать их пересечение.

Первая часть – одно вхождение буквы до точки. Вторая часть – подстрока между точкой и символом **@**.

[A-Z].[a-z]{1, 8}@sun.ru

[A-E].[a-z]+kir@sun.ru

1. **[A-E]** входит в **[A-Z]**, следовательно их пересечение – **[A-E]**

2. **[a-z]+kir** имеет пересечение с **[a-z]{1,8}** в виде **[a-z]{1,5}kir**

Таким образом регулярное выражение, полученное из пересечения двух вышеуказанных будет:

[A-E].[a-z]{1,5}kir@sun.ru

Осталось только посчитать количество строк, попадающих под это регулярное выражение. Это будет количество букв в интервале **[A-E]**, домноженное на количество строк, полученных из **[a-z]{1,5}**. Количество букв в интервале **[a-z]** равно 26, следовательно для каждой возможной длины строки от 1 до 5 включительно посчитаем количество подходящих строк как 26^n , где n равно длине строки.

Получаем $5 * (26 + 26^2 + 26^3 + 26^4 + 26^5) = 61783150$

4. Системы счисления (2 балла)

[Сложная система]

Найдите такие X, Y, Z при которых следующая система будет верной:

$$\begin{cases} A6Z_{16} - 72X_8 = 892_{16} \\ 3XX_4 + 21Y_{16} = 24A_{16} \\ DY0_{16} - 67_8 = DX9_{16} \end{cases}$$

В ответе укажите три числа в десятичной системе счисления через пробел – сначала значение X , затем Y и Z .

Ответ: 1 5 3

Решение:

Приведем систему уравнений к десятичной системе счисления:

$$\begin{cases} 10 * 16^2 + 6 * 16 + Z - 7 * 8^2 - 2 * 8 - X = 8 * 16^2 + 9 * 16 + 2 \\ 3 * 4^2 + X * 4 + X + 2 * 16^2 + 1 * 16 + Y = 2 * 16^2 + 4 * 16 + 10 \\ 13 * 16^2 + Y * 16 + 0 - 6 * 8 - 7 = 13 * 16^2 + X * 16 + 9 \end{cases}$$

Упростим:

$$\begin{cases} Z - X = 2 \\ X * 5 + Y = 10 \\ Y = 4 + X \end{cases}$$

Подставим во второе уравнение Y из третьего уравнения и решим уравнение относительно X , а также выразим Z через X , получим:

$$\begin{cases} Z = 2 + X \\ X = 1 \\ Y = 4 + X \end{cases}$$

Отсюда получаем:

$$\begin{cases} X = 1 \\ Y = 5 \\ Z = 3 \end{cases}$$

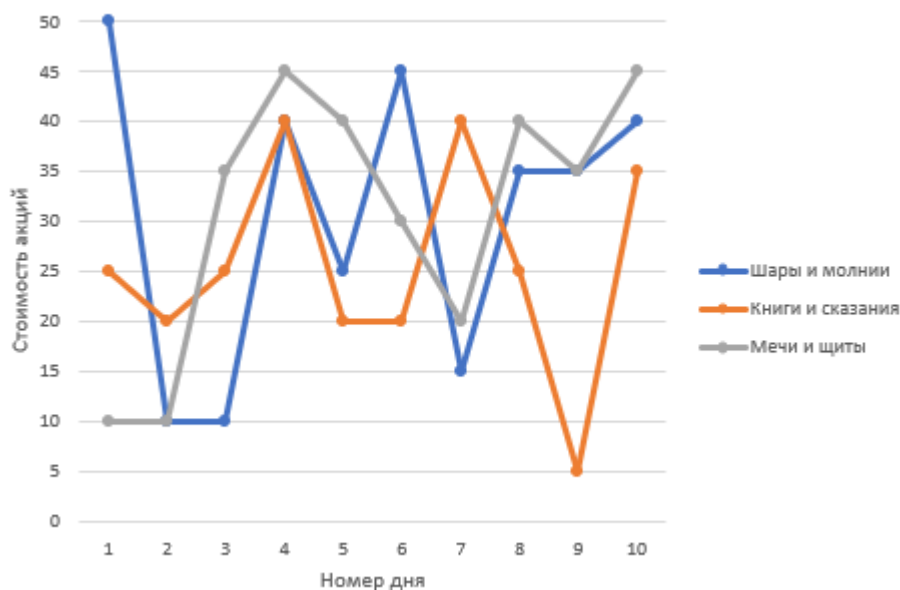
5. Моделирование. Таблицы и графики (1 балл)

[Выгодное вложение]

У девочки Леры есть 50 монет, на эти деньги она хочет купить несколько акций одной из трех магических компаний. Пророк дал Лере листочек с графиком, на котором указаны цены акций в разные дни для каждой из трех компаний. Лера хочет купить на все деньги акции только одной магической компании в один день, а потом продать их все в другой день. Покупка и продажа осуществляются по цене акции в день покупки и день продажи соответственно. Помогите Лере получить максимальную выгоду.

В ответе укажите одно число – максимальную прибыль, которую получит Лера при продаже акций.

График роста акций магических компаний



Примечание: прибыль – разница между продажей и покупкой акций.

Ответ: 300

Решение:

Для решения данной задачи нужно для каждой из трех ломанных, соответствующих стоимостям акций компаний, найти наибольшую выгоду, чтобы затем выбрать самую выгодную компанию из трех.

Чтобы найти наибольшую выгоду от купли-продажи акций одной компании, нужно для каждого дня найти день после исследуемого с наибольшей стоимостью акции.

Тогда получаем самые выгодные решения для каждой компании:

Шары и молнии: купить в 2-ой или 3-ий день, продать в 6-ой день. Разница стоимости (выгода от продажи) между этими днями равна $45-10=35$, а по стоимости в 10 монет Лера может купить только 5 акций. Таким образом, наибольшая выгода от продажи равна $35*5=175$.

Книги и сказания: купить в 9-ый день, продать в 10-ый день. Разница стоимости (выгода от продажи) между этими днями равна $35-5=30$, а по стоимости в 5 монет Лера может купить 10 акций. Таким образом, наибольшая выгода от продажи равна $30*10=300$.

Мечи и щиты: купить в 1-ый или 2-ой день, продать в 4-ый или 10-ый день. Разница стоимости (выгода от продажи) между этими днями равна $45-10=35$, а по стоимости в 10 монет Лера может купить только 5 акций. Таким образом наибольшая выгода от продажи равна $35*5=175$.

Отсюда получаем, что выгоднее всего купить в 9-ый день 10 акций компании Книги и сказания, а продать их в 10-ый день. Выгода равна 300 монет.

6. Кодирование информации. Шифрование (3 балла)

[Коллизия]

Одной из важных функций для работы со строками является хэш-функция. Хэш-функция позволяет преобразовать строку в число, притом единственным образом, а такое число называется хэшем. В идеальном варианте одному хэшу соответствует только одна строка.

Однако это не всегда так. Коллизией хэшей называется такая ситуация, когда для двух разных строк хэш-функция вычисляет одинаковый хэш.

В данной задаче Вы будете работать с полиномиальной хэш-функцией.

Будем вычислять функцию $h(S)$ для исходной строки S длиной n следующим образом:

Сначала для всех i от 0 до n не включительно посчитаем значения $S[i]*p^i$ и сложим их в одно число A . В условиях данной задачи будет считать, что строка S состоит только из латинских прописных букв, а $S[i]$ – номер i -ой буквы строки в алфавите. Нумерация символов строки начинается с нуля, нумерация букв алфавита начинается с нуля.

Вычислим значение функции $h(S) = A \bmod k$, где функция $A \bmod B$ равна остатку от деления A на B .

Для получения различных хэш-функций константы p и k задаются как целые положительные числа.

Вам даны две строки: “had” и “ecc”. Найдите такое минимальное целое положительное p , что при $k=34$ произойдёт коллизия хэшей.

Примечание. Латинский алфавит: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Ответ: 11

Решение:

Давайте, согласно условию, составим для обоих слов функции, задающие хэш:

1. Для слова “had” получится $(7+0*p+3*p^2) \bmod 34$

2. Для слова “ecc” получится $(4+2*p+2*p^2) \bmod 34$

Коллизия произойдёт при условии $(7+0*p+3*p^2) \bmod 34 = (4+2*p+2*p^2) \bmod 34$, то есть остатки от деления на 34 будут равны

Иначе говоря, если разница между $7+0*p+3*p^2$ и $4+2*p+2*p^2$ будет нацело делиться на 34.

Вычтем вторую функцию из первой, получим: $(3*p^2+0*p+7) - (2*p^2+2*p+4) = p^2-2p+3$, данная парабола направлена ветвями вверх и её центр выше оси OX , следовательно разность будет положительна всегда.

Тогда задача сводится к нахождению целого p такого, что $p^2-2p+3 = 34x$, где x – любое целое неотрицательное число.

При $x=0$, решения нет

При $x=1$ и $x=2$, целочисленного решения нет

При $x=3$ ответ равен $p=-11$, $p=11$. Однако, согласно условию, подходит только положительное p , таким образом ответ равен 11.

7. Алгоритмизация и программирование. Анализ алгоритма (2 балла)

[Что такое КМП?]

Вам дан алгоритм:

```

алг Пи функция (арг сим таб S[0:n-1], арг цел n)
нач цел i, j, цел таб P[0:n-1]
    i := 0
    нц пока i < n
        P[i] := 0
        i := i + 1
    кц
    i := 1
    нц пока i < n
        j := P[i - 1]
        нц пока j > 0 И S[i] != S[j]
            j := P[j - 1]
        кц
        если S[i] = S[j]
            j := j + 1
        P[i] = j
        Вывод P[i], '\n'
        i := i + 1
    кц
конец

```

На вход алгоритму подали строку S - "abr#abracadabra" и число n – длина строки S.

Ваша задача – найти, сколько раз было выведено число 3.

В ответе укажите одно число.

Ответ: 2

Решение:

На самом деле данный алгоритм – алгоритм нахождения префикс-функции. Префикс-функция является очень распространенным алгоритмом в задачах со строками, а $P[i]$ – не что иное, как наибольшая длина подстроки, заканчивающейся на индексе i , совпадающей с началом строки. При этом $P[i]$ всегда меньше i .
Пояснение: если $P[i] > 0$, то $S[0 \dots P[i]-1] = S[i-(P[i]-1) \dots i]$

Так же есть алгоритм Кнута-Морриса-Пратта, позволяющий найти количество вхождений подстроки S в строке T. Его суть заключается в том, чтобы применить префикс функцию к строке, образованной склейкой строки S и T через символ, не встречающийся ни в одной из этих двух строк (в данной случае символ '#'). Тогда количество $P[i]$ равных длине строки S будет равно количеству вхождений подстроки S в строке T.

Чтобы решить данную задачу давайте для каждого этапа цикла по i от 1 до n выпишем $P[i]$, так как можно заметить, что на каждом этапе выполнения цикла изменяется только один элемент массива - $P[i]$, и вычисляется он относительно уже посчитанных значений массива P.

$P[0]$ при этом является начальным значением и равен 0. Составим таблицу, где будем записывать все изменения i и j , а так же соответствующие им $S[i]$ и $S[j]$, которые определяют остановку цикла и значение $P[i]$:

i	j	S[i]	S[j]	P[i]
1	0	b	a	0
2	0	r	a	0
3	0	#	a	0
4	0	a	a	1
5	1	b	b	2
6	2	r	r	3
7	3	a	#	
7	0	a	a	1
8	1	c	b	
8	0	c	a	0
9	0	a	a	1
10	1	d	b	
10	0	d	a	0
11	0	a	a	1
12	1	b	b	2
13	2	r	r	3
14	3	a	#	
14	0	a	a	1

Получаем в конце выполнения такой массив P:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
P	0	0	0	0	1	2	3	1	0	1	0	1	2	3	1

В условии задачи спрашивалось количество выведенных чисел 3, их было выведено ровно 2.

8. Алгебра логики (2 балла)

[Потерявшийся оператор]

Вам дана функция от четырех переменных:

$$F(A, B, C, D) = (A \leftrightarrow B) \mid (B \times C) \mid (C \rightarrow D)$$

Где за X скрыта одна из логических операций:

1. $\mid$ (ИЛИ)
2. $\&$ (И)
3. $\leftrightarrow$ (ЭКВИВАЛЕНТНО)
4. XOR (НЕ РАВНО, ИСКЛЮЧАЮЩЕЕ «ИЛИ»: $A \text{ XOR } B = \text{НЕ}(A \leftrightarrow B)$)
5. $\rightarrow$ (СЛЕДОВАНИЕ: $A \rightarrow B = \text{НЕ } A \text{ ИЛИ } B$)
6. $\leftarrow$ (ОБРАТНОЕ СЛЕДОВАНИЕ: $A \leftarrow B = A \text{ ИЛИ НЕ } B$)

Какие операции могут быть скрыты за X такие, что функция F будет принимать значение «истина» ровно для 15 различных наборов значений аргументов (A,B,C,D)?

В ответе перечислите номера операций в порядке возрастания без пробелов и запятых.

Ответ: 2346

Решение:

Давайте для начала составим часть таблицы истинности:

A	B	C	D	$(A \leftrightarrow B)$	$(C \rightarrow D)$
0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	0
1	1	1	1	1	1

Две уже известные операции дают истину для 14 наборов аргументов. Операция $B \times C$ может дать для общей функции F ещё 0, 1 или 2 набора аргументов, для которых функция будет принимать значение истина. Составим таблицу истинности для B и C на наборах (1,1) и (0,1) – это те наборы, которые выделены желтым в предыдущей таблице истинности. Результаты для остальных наборов в данном случае нам не интересны, так как они не изменят количество наборов аргументов, для которых функция F будет истинной.

B	C	$B \mid C$	$B \& C$	$B \leftrightarrow C$	$B \text{ XOR } C$	$B \rightarrow C$	$B \leftarrow C$
0	1	1	0	0	1	1	0
1	1	1	1	1	0	1	1

Чтобы функция F приняла значение истина ровно на 15 наборах аргументов, нужно взять функции из таблицы истинности, которые принимают значение истина ровно 1 раз, это функции 2,3,4 и 6

9. Алгоритмизация и программирование. Формальный исполнитель (2 балла)

[Затейливая магия]

В волшебном городе Строкалесе живут маги, познающие магию строк. Одно из базовых магических заклинаний преобразует строку таким образом, что каждой букве исходной строки справа приписывается зеркальная ей буква, то есть буква, которая стоит на той же позиции в зеркально отраженном алфавите.

Таким образом для буквы **a** зеркальной буквой является буква **z**, а для буквы **f** – **u**.

Так, если применить данное заклинание к строке 'abc', то она преобразуется в строку 'azbyx'.

Ваша задача, как новичка в Строкалесе, определить, какая буква будет стоять на **300-ой** позиции, если к строке 'sobaka' применить заклинание последовательно **7 раз**. Нумерация букв в строке начинается с нуля.

В ответе укажите только букву латинского алфавита.

Ответ: y

Решение:

Для решения задачи давайте попробуем вывести формулу для определения буквы, стоящей на позиции p на шаге m . Промоделируем на примере строки 'abc', и попробуем найти закономерность:

На 0 шаге строка равна 'abc', на первом шаге строка равна 'azbuxc'. То есть исходные символы перемещаются со своей позиции i на позицию $i*2$. А в новой строке все буквы, стоящие на позициях i , не кратных 2, это зеркальная буква к той, что стоит на позиции левее, то есть на позиции $i-1$.

Несложно заметить, что:

Если p не кратно 2, буква на позиции p на шаге m – это зеркальная буква к той букве, которая стоит на позиции $(p-1)/2$ на шаге $m-1$

Если p кратно 2, буква на позиции p на шаге m – это буква, которая стоит на позиции $p/2$ на шаге $m-1$

Давайте пройдемся от 300-ой позиции на 7 шаге до соответствующей позиции на шаге 0 (исходная строка), чтобы определить ответ на поставленную задачу. Проходить будем согласно правилам, которые определили выше.

Примечание: зеркальная буква к зеркальной букве – исходная буква.

7 шаг: 300-ая буква

6 шаг: 150-ая буква

5 шаг: 75-ая буква зеркальная к 5 шагу

4 шаг: 37-ая буква зеркальная к 4 шагу

3 шаг: 18-ая буква

2 шаг: 9-ая буква зеркальная к 1 шагу

1 шаг: 4-ая буква

0 шаг: 2 буква

При проходе от 7 шага к 0-ому было три операции «отзеркаливания» буквы, следовательно буква на 300-ой позиции на 7 шаге – зеркальная буква к букве, стоящей на второй позиции в исходной строке. Зеркальная буква для буквы b – y .

10. Комбинаторика (1 балл)

[Опять DnD?]

Как-то раз группа друзей собрались поиграть в настольную приключенческую игру Demons&Dwarfs. В этой игре, чтобы сделать какое-либо действие, нужно бросать кубики.

В один из моментов приключения одному из игроков на пути встретился циклоп, и чтобы пройти дальше – нужно было победить его.

Чтобы проверить, попал ли удар по врагу, нужно кинуть D20 кубик; считается, что игрок попал по врагу, если число, описывающее количество **брони врага**, не больше числа, выпавшего на кубике. Затем, если удар попал по врагу, то нужно кинуть 4 кубика D4 на атаку, и если сумма на выпавших гранях кубиков не меньше, чем число, описывающее количество **жизней врага**, то игрок победил. Таким образом, в зависимости от первого броска, может получиться последовательность из одного или пяти бросков. Даже если меньшего числа кубиков D4 хватает для победы, кинуть нужно все кубики. У кубиков D20 и D4 грани содержат неповторяющиеся числа от 1 до 20 и от 1 до 4 соответственно.

У Циклопа **14 брони** и **9 жизней**.

Сколько различных последовательностей бросков таких, что по итогу бросков кубиков, игрок победит циклопа? В рамках данной задачи полагается, что кубики на урон бросаются одновременно и варианты с одинаковым набором чисел, выпавших на гранях, считаются за один вариант. Например, если на кубиках D4 выпали числа 1,2,3,3, то любые варианты перестановок этих чисел между кубиками не считаются за новый вариант.

Пример: допустим у врага характеристики: 10 брони и 6 жизней

Игрок бросает кубик D20 на проверку попадания, выпадает 12 (что больше 10). Следовательно, можно кидать кубики D4 на урон. Затем он кидает 4 кубика D4 на урон, если выпадет сумма больше или равная 6, то он побеждает.

В ответе укажите одно число – количество последовательностей бросков.

Ответ: 168

Решение:

Чтобы посчитать количество различных последовательностей бросков, нужно понимать, что кубики на проверку попадания по врагу и кубики на урон бросаются независимо. Таким образом, общее количество различных последовательностей бросков для того, чтобы победить Циклопа – перемножение количества возможных бросков кубика D20, чтобы выпало не менее 14, и количества возможных наборов бросков четырех кубиков D4 с суммой не меньше 9.

Количество возможных бросков кубика D20, чтобы выпало не менее 14, равно 7.

Посчитаем количество наборов бросков четырех кубиков D4 с суммой не меньше 9. Возможные суммы: 9,10,11,12,13,14,15 и 16

Для 9: (1,1,3,4), (1,2,2,4), (1,2,3,3), (2,2,2,3)

Для 10: (1,1,4,4), (1,2,3,4), (1,3,3,3), (2,2,2,4), (2,2,3,3)

Для 11: (1,2,4,4), (1,3,3,4), (2,2,3,4), (2,3,3,3)

Для 12: (1,3,4,4), (2,2,3,4), (2,3,3,4), (3,3,3,3)

Для 13: (1,4,4,4), (2,3,4,4), (3,3,3,4)

Для 14: (2,4,4,4), (3,3,4,4)

Для 15: (3,4,4,4)

Для 16: (4,4,4,4)

Итого 24 варианта.

Таким образом количество различных последовательностей бросков для того, чтобы победить Циклопа равно $7*24=168$

Отборочный этап. Первый тур (приведен один из вариантов заданий)

1. Теоретические основы информатики, одиночный выбор. 1 балл [UML]

Какое из нижеперечисленных названий не соответствует ни одному типу диаграмм UML2.1?

1. Диаграмма классов (Class diagram)
2. Диаграмма компонентов (Component diagram)
3. Диаграмма переходов (Switch-over diagram)
4. Диаграмма последовательности (Sequence diagram)
5. Диаграмма деятельности (Activity diagram)

В ответе укажите только число, соответствующее правильному ответу.

Ответ: 3

2. Теоретические основы информатики, одиночный выбор. 1 балл [BidData]

Как называется операция в модели распределенных вычисления MapReduce, задача которой заключается в перераспределении данных по ключам между вычислительными нодами?

1. Map
2. Reduce
3. Rework
4. Shuffle
5. Distribution

Ответ: 4

3. Комбинаторика. (1 балл) [D20 куб]

Как-то раз группа друзей собрались поиграть в настольную приключенческую игру Demons&Dwarfs. В этой игре, чтобы сделать какое-либо действие, нужно бросать кубики. В сложной ситуации одному из мальчиков нужно было атаковать сразу двух(!) врагов, от этой атаки зависел успех всей команды. Чтобы проверить, попал ли удар по врагам, нужно кинуть D20 кубик; считается, что игрок попал по всем врагам, у которых число, описывающее количество брони врага, не больше числа, выпавшего на кубике. Затем, если удар попал по врагу, то нужно кидать D8 кубик на атаку против каждого врага, и если на соответствующем кубике выпало не меньше, чем число, описывающее количество жизней врага, то игрок убил его. Таким образом, в зависимости от первого броска, может получиться последовательность из одного, двух или трех бросков, но только последовательностью из трех бросков можно убить обоих противников. У кубиков D20 и D8 грани содержат неповторяющиеся числа от 1 до 20 и от 1 до 8 соответственно.

Первый босс – огромный орк, у него 6 брони и 6 жизней.

Второй босс – ловкий гоблин, у него 14 брони и 4 жизни.

Сколько различных последовательностей бросков таких, что по итогу бросков кубиков, мальчик убьёт обоих врагов? В рамках данной задачи очередность бросков на урон (если можно атаковать обоих врагов) определяется номером босса.

Пример: допустим у врагов характеристики:

1. 10 брони и 4 жизни
2. 12 брони и 2 жизни

Игрок бросает кубик D20 на проверку попадания, выпадает 12. Следовательно по обоим врагам он попадает и должен кинуть D8 на урон против первого босса.

Затем он кидает D8 на урон по первому врагу, если выпадет ≥ 4 , то он убивает первого босса и должен кинуть кубик против второго босса. Во втором броске он должен выбросить кубик на ≥ 2 , чтобы убить второго босса. Если бы при броске урона на первого босса выпало < 4 , то броски закончились, так как не получится убить обоих.

В ответе укажите одно число – количество последовательностей бросков.

Ответ: 105

4. Шифрование информации. 2 балла [Каменный шифр]

Петя и Вася очень любят шпионскую тематику. Они придумали новый способ шифрования с помощью камней. Они выбрали 8 цветов и пронумеровали их:

0	1	2	3	4	5	6	7
черный	красный	оранжевый	жёлтый	зелёный	голубой	синий	фиолетовый

Шифр работает следующий образом:

1. Один из мальчиков составляет сообщение, состоящее из букв английского алфавита и цифр.
2. Это сообщение является числом в 36-ричной системе счисления. Как известно, при основаниях систем счисления, больших 10, для цифр больше 9 используются последовательно заглавные буквы латинского алфавита.
3. Каждый символ сообщения – 36-ричная цифра – переводится в двоичную систему счисления и записывается в виде 6 двоичных разрядов (при необходимости добавляются незначащие нули).
Каждый камень имеет номер, который так же представляется в двоичном виде, например синий камень (номер 6) это 110_2 .

Таким образом, каждый символ сообщения может быть представлен двумя камнями.

Пример: символ “F” считаем 36-ричным числом. $F_{36}=001111_2$. Разделим на две части: 001 и 111. Первой части соответствует красный камень, второй – фиолетовый. Таким образом символ “F” в сообщении кодируется как



4. Каждый символ сообщения шифруется как в пункте 3, и зашифрованное сообщение выкладывается слева-направо.

Петя составил сообщение, зашифровал его описанным выше способом и разложил зашифрованное сообщение камнями. Вася в этот момент сидит напротив (соответственно видит порядок камней перевернутым). Помогите Васе расшифровать сообщение, которое он видит:



В ответе укажите исходное сообщение, используйте заглавные буквы латинского алфавита и цифры.

Ответ: TEAM

5. Системы счисления. 2 балла

[Сложное уравнение]

Найдите количество пар цифр X, Y, при которых будет выполняться следующее равенство:

$$3XD_{16} + YZ_8 = 100020_4$$

В ответе укажите только количество пар.

Ответ: 4

6. Кодирование информации. Видео. 3 балла

[ТокТик]

Девочка Лера очень любит сидеть в свободное время на новой платформе с короткими видео под названием ТокТик. Периодически ей видео нравятся настолько, что она хочет поделиться ими с друзьями. И в один из дней Лера зашла в свою галерею и задалась вопросом, сколько существует вариантов выбрать несколько видео так, чтобы они успели отправиться за 10 минут?

Скорость интернета Леры равна 5760 Кбит/с.

В рамках данной задачи считайте, что время отправки нескольких файлов считается как сумма времени, потраченного на отправку каждого отдельного файла.

В ТокТик видео представляют из себя последовательный набор кадров без сжатия, а количество кадров в секунду всегда равно 30. Каждый кадр – последовательность кодов пикселей без дополнительной информации. Так же вам про каждое видео известно его ширина и высота (в пикселях), а также глубина цвета и продолжительность в секундах.

Номер видео	Высота, px	Ширина, px	Глубина цвета, бит	Продолжительность, с
1	860	540	8	20
2	640	360	8	10
3	430	240	16	20
4	640	360	16	30
5	860	540	8	20

В ответе укажите одно число - количество вариантов. Варианты с одинаковым набором видео, но с разным порядком, считаются одним вариантом.

Примечание: 1Кбит/с = 1000 бит/с.

Ответ: 10

7. Количество вариантов. 2 балла

[Лифт сломан или...]

Вам была дана задача протестировать лифт нового поколения в новом 50-этажном бизнес-центре. На обычном лифте вас подняли на 9 этаж и сопроводили до нового лифта. В этом лифте нет кнопок с номерами этажей, есть только кнопки «ВВЕРХ» и «ВНИЗ». Так же вы замечаете рядом инструкцию:

Пусть x – номер текущего этажа.

Если x – чётное и нажата кнопка «ВВЕРХ», то лифт привезёт Вас на этаж с номером $2x-1$

Если x – нечётное и нажата кнопка «ВВЕРХ», то лифт привезёт Вас на этаж с номером $2x$

Если x – чётное и нажата кнопка «ВНИЗ», то лифт привезёт Вас на этаж с номером $x/2$

Если x – нечётное и нажата кнопка «ВНИЗ», то лифт привезёт Вас на этаж с номером $(x+1)/2$

Если номер этажа, на который должен отправиться лифт, отсутствует – лифт не сдвинется с места.

Нумерация этажей бизнес-центра начинается с 1.

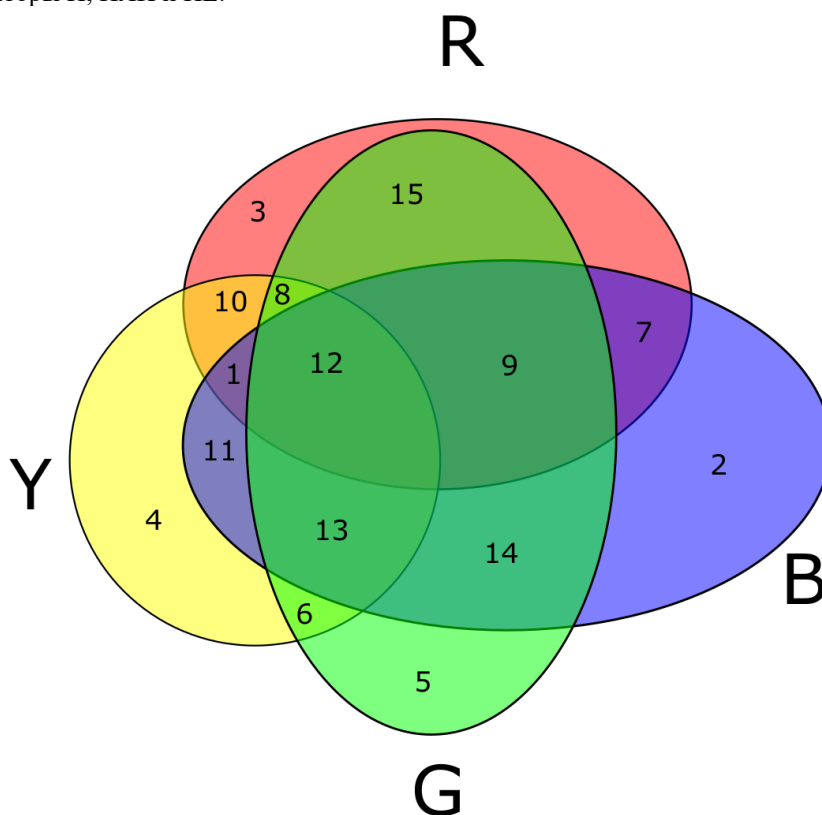
Вам можно составлять любые последовательности нажатия кнопок, и в любой момент времени можете вернуться на стартовый этаж. Сколько различных этажей можно посетить из всех возможных таких последовательностей?

Ответ: 14

8. Основы логики. 1 балл

[Кусочки пластика]

Мальчик Илья нашёл четыре полупрозрачных кусочка пластика разных цветов: R, G, B и Y и расположил их, как показано на рисунке. Как видно из рисунка, можно выделить 15 непересекающихся зон, в каждой из которых свой цвет и номер и присутствует один кусочек или наложение из нескольких кусочков. Тогда мы можем описать ту или иную зону, используя названия кусочков и операторы И, ИЛИ и НЕ.



Вам дано описание некоторой области:

$$(R \text{ или } Y) \text{ и } G \text{ и } B \text{ или } R \text{ и } Y \text{ и } (B \text{ или } G)$$

Укажите все зоны, попадающие в эту область. В ответе укажите их через запятую, без пробелов, в возрастающем порядке номеров зон.

Ответ: 1,8,9,12,13

9. Алгебра логики. 2 балла

[Скрытая функция]

Вам дана функция от трех переменных:

$$F(A, B, C) = (A \& B) \mid (B \text{ X } C)$$

Где за X скрыта одна из логических операций:

1. $\mid$ (ИЛИ)
2. $\&$ (И)
3. $\lt \gt$ (ЭКВИВАЛЕНТНО)
4. $\wedge$ (НЕ РАВНО)

5. $\rightarrow$ (СЛЕДОВАНИЕ: $A \rightarrow B = \text{НЕ } A \text{ ИЛИ } B$)
6. $\leftarrow$ (ОБРАТНОЕ СЛЕДОВАНИЕ: $A \leftarrow B = A \text{ ИЛИ НЕ } B$)

Какие операции могут быть скрыты за **X** такие, что функция F будет принимать значение «истина» для более чем 5 различных наборов значений аргументов (A,B,C)?

В ответе перечислите номера функций в порядке возрастания без пробелов и запятых.

Ответ: 156

10. Количество вариантов. 3 балла

[Камешки]

Мальчики Андрей и Илья придумали игру с камешками. Суть игры заключается в том, чтобы набрать кучку с определенным количеством камешков в ней, добавляя камешки только определенными операциями:

- 1) Добавить в кучку 3 камешка, если сейчас в кучке число камешков не кратно пяти
- 2) Добавить в кучку 1 камешек, если сейчас в кучке нечетное число камешков
- 3) Увеличить количество камешков в 1.5 раза, если сейчас в кучке четное число камешков

Играет один человек и делает последовательные операции. Игра начинается с кучки из 1 камешка.

Сколько различных последовательностей операций есть таких, что в кучке наберется 22 камешка?

Ответ: 38

Отборочный этап. Второй тур (приведен один из вариантов заданий)

1. Теоретические основы информатики, множественный выбор. [1 балл] [Система контроля версий]

Используя поиск информации в глобальной сети Интернет определите, какие из предложенных команд git существуют? Выберите все верные варианты ответа.

1. git reset
2. git status
3. git delete
4. git sort
5. git change all

Ответ: 12

2. Теоретические основы информатики, единичный выбор. [1 балл] [Средства разработки]

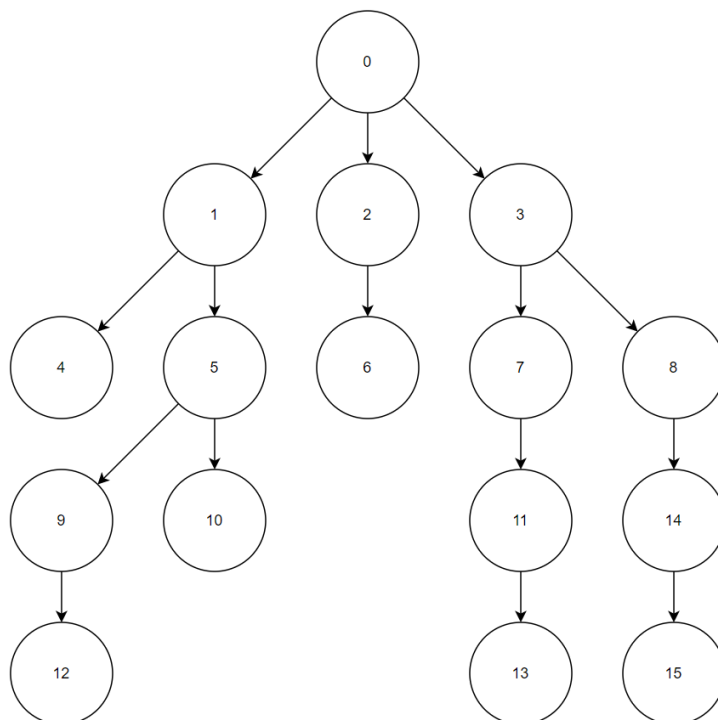
Используя поиск информации в глобальной сети Интернет определите, что из предложенного ниже не является IDE? Выберите верный вариант ответа.

1. Visual Studio
2. VirtualBox
3. PyCharm
4. Clion
5. Android Studio

Ответ: 2

3. Анализ кода. [2 балла] [Деревья]

Вам дано графическое представление дерева:



Деревом называется граф иерархической структуры, дерево не содержит циклов, между двумя вершинами может быть только один путь.

Вершина дерева – уникальный объект дерева.

Между двумя вершинами может быть только одна связь.

Если из вершины А выходит связь в вершину В, то вершина А является родителем вершины В.

Корень – главная вершина дерева, имеет номер 0.

Глубина вершины – количество связей в пути от корня до вершины.

Для хранения информации о дереве решили использовать два массива:
 Массив d имеет длину 16 и d[i] – глубина вершины с номером i. Например d[11]=3.
 Массив p имеет длину 16 и p[i] – родитель вершины с номером i. Например p[11]=7.
 Корень имеет глубину 0 и является родителем самого себя. Таким образом p[0]=0, d[0]=0.

Вам дан алгоритм. Так же Вам известно, что a и b – номера вершин дерева.

```

Алг
Нач
| цел: a = 11, b = 15
|
| если d[a] < d[b] то
| | t = a
| | a = b
| | b = t
| всё
|
| нц пока d[a] > d[b]
| | a = p[a]
| кц
|
| нц пока a != b
| | a = p[a]
| | b = p[b]
| кц
|
| Вывод a
Кон
    
```

Укажите, что выведет представленный выше алгоритм, зная структуру дерева.

Ответ: 3

4. Электронные таблицы, моделирование. [2 балла] [Длинные формулы]

Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D
1	=СУММ(B2:V17)	0	=B1+1	
2	0	=ОСТАТ(		
3	=A2+1	ЕСЛИ(И(2 <= B\$1; B\$1 <= 10; 4 <= \$A2; \$A2 <= 7); 1; 0) +		
4		ЕСЛИ(И(9 <= B\$1; B\$1 <= 16; 7 <= \$A2; \$A2 <= 12); 1; 0) +		
5		ЕСЛИ(И(\$X\$1 <= B\$1; B\$1 <= \$X\$1 + \$Y\$1; \$X\$1 <= \$A2; \$A2 <= \$X\$1 + \$Z\$1); 1; 0);2)		
6				

Ячейку C1 скопировали во все ячейки диапазона D1:V1, ячейку A3 скопировали во все ячейки диапазона A4:A17. Ячейку B2 скопировали во все ячейки диапазона B2:V17.

Значение ячейки Y1 равно 4, ячейки Z1 равно 2.

После перевода таблицы в режим отображения чисел значение ячейки A1 стало равно 75.

Найдите все возможные целые неотрицательные значения ячейки X1, меньше 12. В ответе укажите все варианты ответа в порядке возрастания через запятую без пробелов.

Ответ: 3,6,11

5. Моделирование. [3 балла] [Эффективное использование трафика]

Вам дана сеть с распределенной передачей данных. В сети есть:

- 1) Источники данных, способные отсылать в сеть 120 Мбит данных в секунду каждый. К источнику может быть подключено только одно сетевое подключение.
- 2) Потребители данных, способные принимать 45 Мбит данных в секунду. В потребитель может входить только одно сетевое подключение.
- 3) Сетевые подключения, пропускная способность которых не более 200 Мбит/с для каждого.
- 4) Коммутаторы, способные соединить до 4 сетевых подключений и перераспределить передаваемые данные. В коммутаторе складываются все входящие потоки данных и распределяются поровну между выходящими сетевыми соединениями. У самих коммутаторов нет ограничения на пропускную способность.

В данной сети данные идут строго от источников к потребителям. Если до потребителя дошел поток данных, превосходящий его возможности приёма, избыточные данные просто потеряются.

Вам нужно сконструировать минимальную (по количеству источников, потребителей и коммутаторов) схему, в которой весь поток данных от источников будет доставляться в полном объёме потребителям без потерь, и все потребители будут получать максимальные потоки данных, которые они могут принять. В ответе укажите количество источников данных, потребителей данных и коммутаторов через запятую без пробелов в указанном порядке.

Ответ считается минимальным, если нельзя получить меньшую сумму количества источников данных, потребителей данных и коммутаторов так, чтобы все данные от источников без потерь потреблялись потребителями и все потребители получали бы максимально возможные для них потоки данных.

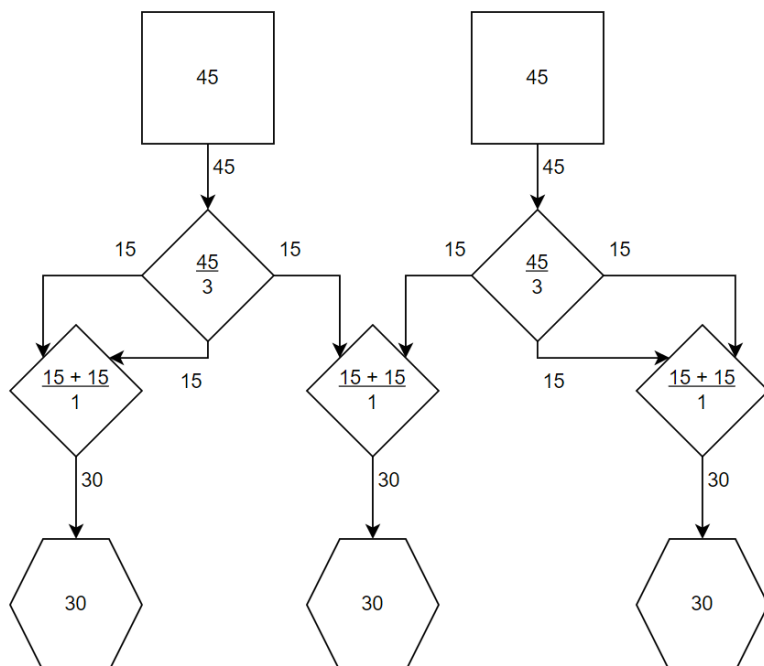
Примеры:

Квадраты – источники данных

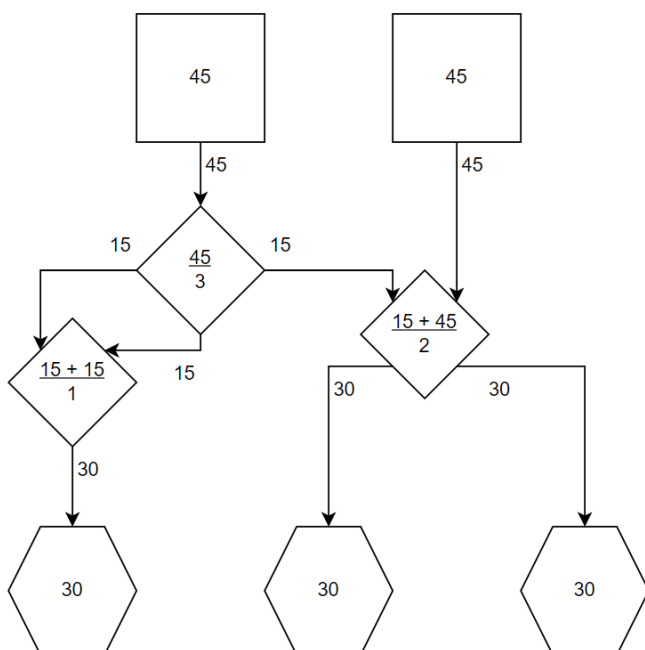
Ромбы – коммутаторы

Шестиугольники – потребители данных

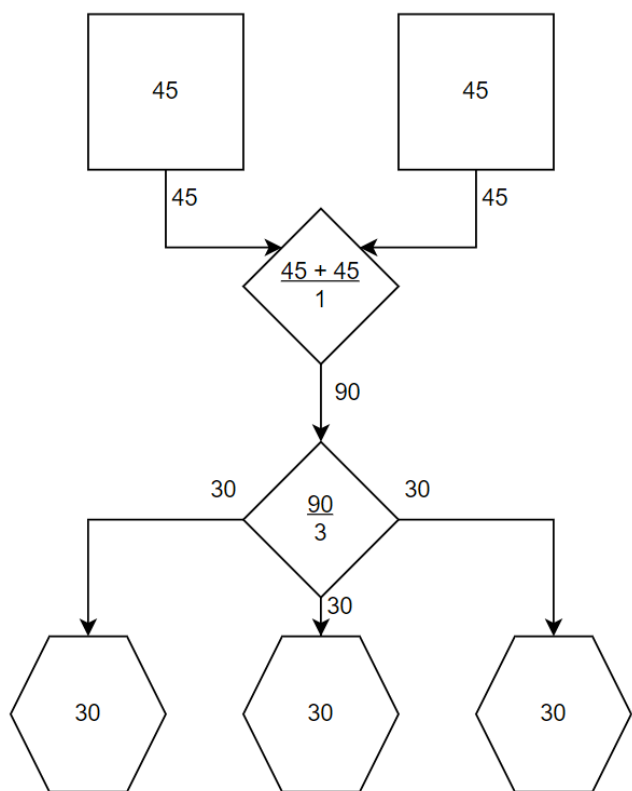
В примерах источники данных производят 45 Мбит в секунду, а потребители потребляют 30 Мбит в секунду. На всех коммутаторах представлены расчеты выходящих потоков данных: в числителе сумма входящих потоков, в знаменателе количество выходящих потоков.



Если пропускная способность сетевого соединения равна 60 Мбит/с, то одна из схем сети может выглядеть как изображено на рисунке слева. Но данная схема не является минимальным ответом.



Если пропускная способность сетевого соединения равна 60 Мбит/с, то минимальный ответ может быть получен из схемы, изображенной слева. А ответ будет описан как 2,3,3



Если же пропускная способность сетевого соединения будет равно 90 Мбит/с, то схема, представленная для сетевого соединения в 60 Мбит/с уже не будет являться минимальным ответом, так как можно уменьшить число коммутаторов как представлено на схеме слева.

Ответ: 3,8,7

6. Моделирование по графу. [3 балла]

[Лабиринт]

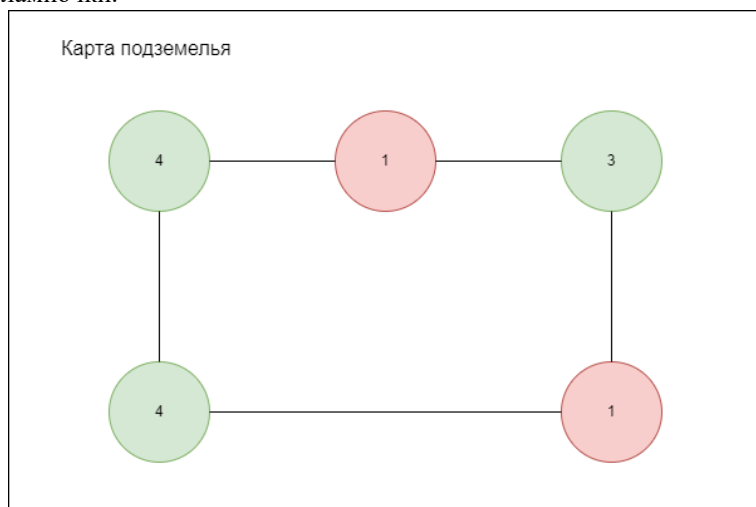
Вася попал в подземелье, которое состоит из комнат и туннелей между ними. В каждой комнате находится по 4 лампочки. В подземелье есть два типа комнат – красные и зеленые. В зеленых комнатах есть кнопка для включения лампочек, а в красных - нет.

С помощью нажатия на кнопку можно включать лампочки по следующему правилу: одно нажатие на кнопку включает по одной лампочке в текущей комнате и во всех соседних комнатах, если в какой-то комнате уже горят 4 лампочки, то при попытке включить еще одну лампочку в этой комнате потухнут 3 лампочки и останется гореть одна.

Вася может нажимать на кнопки в любом порядке, а также беспрепятственно передвигаться по подземелью и заходить в любую комнату.

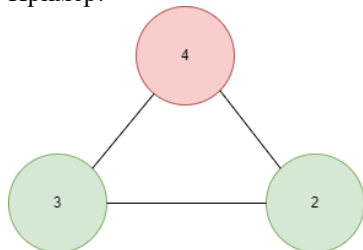
Вам дан план подземелья. Комнаты изображены на плане как круги, а число внутри круга – количество горящих лампочек на момент попадания Васи в подземелье. Помогите Васе узнать минимальное количество нажатий на кнопки, чтобы в каждой комнате горело ровно по 4 лампочки.

В ответе укажите одно число – минимальное количество нажатий кнопок, чтобы в каждой комнате было зажжено 4 лампочки.



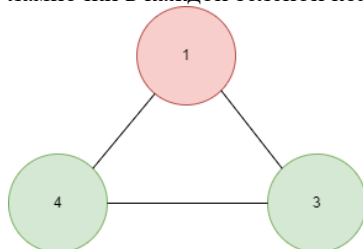
Соседней комнатой называется любая комната, которая соединена туннелем с текущей комнатой.

Пример:



Вася может нажимать кнопки только в зеленых комнатах.

Если Вася нажмет на кнопку в комнате с 3 лампочки, то потухнет 3 лампочки в красной комнате, и загорится по одной лампочки в каждой зеленой комнате.



Ответ: 5

7. Маски. [2 балла]

[Регулярные выражения]

Для задания регулярных выражений приняты следующие обозначения:

c	Любой неспециальный символ c соответствует самому себе. Специальными символами будем считать только символы [,], {, }, *, +, -, ? – эти символы не могут по условию данной задачи встретиться в тексте.
[...]	Любой символ из ... ; допустимы диапазоны типа: • a-z (последовательно идущие символы в алфавите), • A-Z (последовательно идущие символы в алфавите), • 0-9 (последовательно идущие цифры). Диапазоны могут быть указаны друг за другом.
r*	Ноль или более вхождений символа r , может применяться и для диапазонов. Например, a* означает ноль или более вхождений символа a .
r+	Одно или более вхождений символа r , может применяться и для диапазонов, например [a-z]+ означает одно или более вхождений символов диапазона a-z в любом порядке.
r1r2	За символом или диапазоном r1 следует символ или диапазон r2 .
{n}	Число вхождений n предыдущего выражения. Например, выражение [a-z]{3} соответствует подстроке из трех латинских букв.
{n, m}	Число вхождений от n до m предыдущего выражения. Например, выражение [a-z]{3, 5} соответствует подстроке из не менее трех и не более пяти латинских букв.

Дан исходный текст.

She sells seashells on the seashore.

The shells she sells are seashells, I am sure.

Then, I am sure she sells seashore shells.

Вам дано регулярное выражение **s#+[a-z]+#**. Части данного регулярного выражения заменили на символы **#**. За одним символом **#** скрыто ровно одно выражение из предложенных ниже. Каждое выражение можно использовать только один раз.

1. **[A-Z]**
2. **l+**
3. **l***
4. **s**
5. **e**
6. **[a-z]**

Помогите восстановить исходное регулярное выражение, чтобы с помощью него можно было найти только выделенные девять фрагментов текста.

She **sells seashells** on the **seashore**.

The **shells** she **sells** are **seashells**, I am sure.

Then, I am sure she **sells seashore shells**.

В ответе напишите номера выражений без пробелов и запятых, в том порядке, в котором они должны находиться в исходном регулярном выражении.

Ответ: 634

8. Моделирование по строке. [2 балла]

[Алгоритм с шифром]

Вам дана исходная строка $s = \text{'CUCUMBER'}$ и число $k = 2$. Строку изменяют по следующему алгоритму:

1. Создают копию имеющейся строки s .
2. Над каждой буквой копии производят следующие операции:
 - а. Пусть x – номер этой буквы в алфавите. Если x кратен $k \% 10$, то заменить эту букву на букву с номером $(x + k) \% 26$. Нумерация букв в алфавите начинается с нуля.
 - б. Иначе перейти к следующей букве.
3. К строке s в конец дописывается измененная копия.
4. Число $k = k * 2 - 1$
5. Если $k \leq 50$, то перейти на шаг 1, иначе завершить алгоритм.

Пояснение: $a \% b$ – остаток от целочисленного деления числа a на b

Вам необходимо найти самую частую букву в строке после выполнения алгоритма

В ответе укажите искомую букву и количество её повторений в строке.

Пример ответа: A15

Ответ: E96

9. Анализ таблиц. [1 балл]

[Стажировка]

Работодатель составил базу данных кандидатов, которые писали вступительные испытания на стажировку.

Данные о кандидатах:

- Id кандидата – уникальный номер кандидата.
- ФИО – фамилия, имя, отчество кандидата. Гарантируется, что ФИО уникальны.
- Курс – номер курса, на котором сейчас учится кандидат.
- Университет – название университета, в котором обучается кандидат.

Данные о прохождении вступительных испытаний:

- Id кандидата – уникальный номер кандидата.
- Направление стажировки – название стажировки, на которую подавался кандидат.
- Баллы – количество баллов, которое набрал кандидат за вступительные испытания.
- Наличие мотивационного письма – может принимать только значения «0» или «1». «0» означает, что у данного кандидата нет мотивационного письма, «1» – мотивационное письмо есть.

Данные об успеваемости кандидата в ВУЗе.

- ФИО – фамилия, имя, отчество кандидата. Гарантируется, что ФИО уникальны.
- Университет – название университета, в котором обучается кандидат.
- Средний балл – это средний балл по дисциплинам последнего семестра в ВУЗе.

Таблицы данных:

Id кандидата	ФИО	Курс	Университет
885845	Сысоев Клим Олегович	2	Синергия
249708	Кулагина Нелли Дмитриевна	1	МГУ
701911	Чернов Марк Германович	3	ИТМО
750005	Котова Оксана Глебовна	2	ВШЭ
627817	Корнилов Николай Мэлорович	4	ИТМО
429364	Быкова Лиза Робертовна	5	ВШЭ
316489	Белоусов Артур Владленович	1	Синергия
685916	Журавлёв Лука Петрович	1	МФТИ
351838	Анисимов Станислав Яковлевич	4	УРФУ
862408	Миронов Мирон Якунович	1	ЮУрГУ
160364	Казаков Ярослав Дмитриевич	2	ЮУрГУ
458412	Пономарёва Надежда Агафоновна	1	МГУ
627999	Карпов Май Семенович	1	Синергия

151094	Терентьева Санта Ивановна	5	МФТИ
605858	Капустина Лили Федотовна	3	ИТМО
157016	Миронова Лилу Артемовна	3	ВШЭ
417320	Белозёрова Татьяна Тимуровна	3	УРФУ
269395	Стрелкова Ираида Аристарховна	1	ИТМО
436432	Авдеев Вальтер Анатольевич	2	Синергия
517899	Михеев Людвиг Ефимович	5	Синергия
895255	Савина Аэлиа Германовна	2	ВШЭ
248665	Зиновьева Наталья Платоновна	5	УРФУ
460323	Исаева Анна Якововна	5	УРФУ
131636	Гусев Людвиг Владленович	3	МГУ
378539	Киселёв Велор Онисимович	5	МФТИ

Id кандидата	Направление стажировки	Баллы	Наличие мотивационного письма
862408	Аналитика	4	1
701911	Аналитика	7	1
429364	Аналитика	6	1
316489	Аналитика	4	0
248665	Аналитика	25	0
157016	Аналитика	0	1
750005	Дизайн	18	1
627999	Дизайн	6	0
458412	Дизайн	12	1
436432	Дизайн	24	1
151094	Дизайн	4	0
131636	Дизайн	20	0
685916	Разработка	22	0
460323	Разработка	12	1
417320	Разработка	19	1
378539	Разработка	12	1
269395	Разработка	15	0
249708	Разработка	3	1
160364	Разработка	16	0
895255	Тестирование	8	1
885845	Тестирование	8	0
627817	Тестирование	25	0
605858	Тестирование	3	1
517899	Тестирование	18	1
351838	Тестирование	11	1

ФИО	Университет	Средний балл
Быкова Лиза Робертовна	ВШЭ	4,16
Котова Оксана Глебовна	ВШЭ	4,18
Миронова Лилу Артемовна	ВШЭ	3,93
Савина Аэлиа Германовна	ВШЭ	3,37
Капустина Лили Федотовна	ИТМО	4,45
Корнилов Николай Мэлорович	ИТМО	3,63
Стрелкова Ираида Аристарховна	ИТМО	3,47
Чернов Марк Германович	ИТМО	3,76

Гусев Людвиг Владленович	МГУ	3,91
Кулагина Нелли Дмитриевна	МГУ	4,03
Пономарёва Надежда Агафоновна	МГУ	3,89
Журавлёв Лука Петрович	МФТИ	3,37
Киселёв Велор Онисимович	МФТИ	4,72
Терентьева Санта Ивановна	МФТИ	3,19
Авдеев Вальтер Анатольевич	Синергия	4,81
Белоусов Артур Владленович	Синергия	3,59
Карпов Май Семенович	Синергия	3,42
Михеев Людвиг Ефимович	Синергия	4,48
Сысоев Клим Олегович	Синергия	4,54
Анисимов Станислав Яковлевич	УРФУ	3,51
Белозёрова Татьяна Тимуровна	УРФУ	3,36
Зиновьева Наталья Платоновна	УРФУ	3,15
Исаева Анна Якововна	УРФУ	4,64
Казаков Ярослав Дмитриевич	ЮУрГУ	4,42
Миронов Мирон Якунович	ЮУрГУ	3,19

Сколько кандидатов из топ-10 по результатам вступительного испытания имеют мотивационное письмо и средний балл больше 4,5?

Топ-10 по результатам вступительного испытания считаются первые 10 кандидатов, которые набрали наибольшее количество баллов.

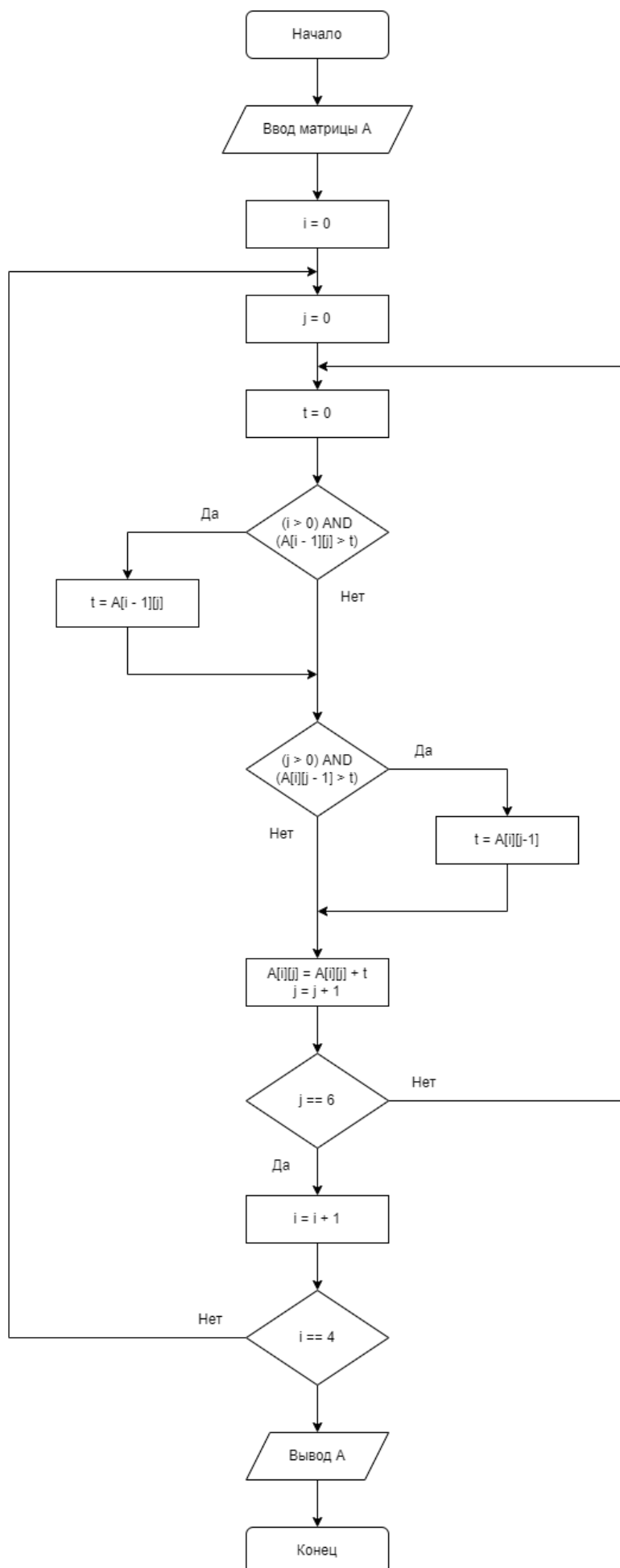
Ответ: 1

10. Блок-схема. [3 балла]

[Двумерный массив]

11. Для представленного ниже алгоритма известно, что на вход ему подали двумерный массив А. Нумерация элементов массива начинается с [0,0]. При обращении к элементам двумерного массива первая координата отвечает за номер строки, вторая – номер столбца. Значения массива А представлены ниже в виде таблицы:

3	57	40	16	6	77
84	55	98	36	76	51
62	66	36	11	76	70
1	94	35	5	18	82



Найдите, чему станет равно $A[3][5]$ после окончания работы алгоритма.

Ответ: 580