

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

---

**Вариант 1**

**Часть 1**

**Задача 1**

**Оцифровка**

Работник архива отсканировал 50 старых фотографий на цветном сканере. После этого объединил в один документ (презентацию) и сделал подписи к каждой фотографии. Общий размер документа стал равен 102584 КиБ.

Каждая страница имеет разрешение 1366 на 768 пикселей и глубину цвета 16 бит.

Определите сколько бит занимает один символ текста, если известно, было напечатано 3072 символа, объем служебной информации в презентации - 128 КиБ.

*Примечание: 1 КиБ = 1024 байт*

**Пример решения:**

1.

```
#!/usr/bin/python3
```

```
#Разрешение отсканированных страниц
```

```
x = 1366
```

```
y = 768
```

```
#Глубина цвета
```

```
deep = 16
```

```
#Количество страниц
```

```
n = 50
```

```
#Размер документа без служебной информации
```

```
N = 102584 - 128
```

```
#Кол-во символов
```

```
text = 3072
```

```
print((N * 1024 * 8 - x * y * deep * n) // text)
```

**Ответ:**

16

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

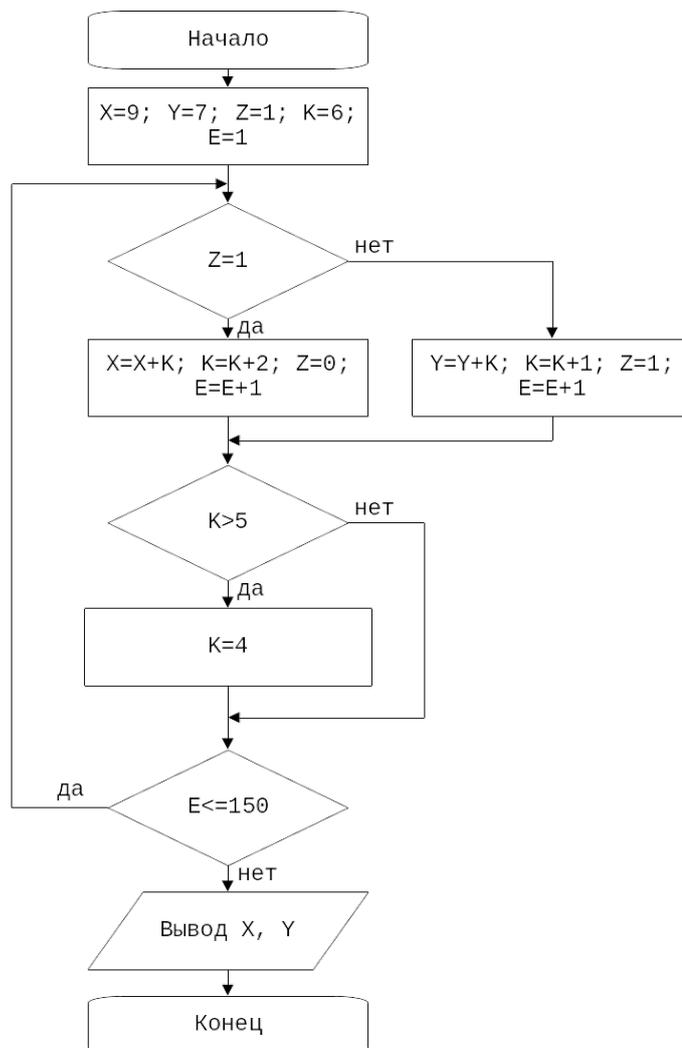
---

**Задача 2**

**Алгоритм**

Дана блок-схема алгоритма.

В ответ запишите модуль разницы чисел  $X$  и  $Y$  после завершения работы алгоритма.



**Пример решения:**

```
#!/usr/bin/python3
```

```
x = 9
```

```
y = 7
```

```
z = 1
```

```
k = 6
```

```
e = 1
```

```
while e <= 150:
```

```
    if z == 1:
```

```
        x += k
```

```
        k += 2
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС

---

```
z = 0
e += 1
else:
    y += k
    k += 1
    z = 1
    e += 1
if k > 5:
    k = 4

print(abs(x - y))
```

**Ответ:**

78

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

---

### Задача 3

#### Равенство

Определите, при каком минимальном значении  $N$  сумма цифр в двоичном представлении числа  $P$  будет равна 6?

$$1N1_4 + 1N1_8 + 1N1_{16} = P.$$

*Примечание: При  $N=0$ , выражение будет иметь вид:  $1014 + 1018 + 10116 = P$*

#### Пример решения:

```
#!/usr/bin/python3
```

```
n = 0
```

```
while 1:
```

```
    P = int(f'1{n}1', 4) + int(f'1{n}1', 8) + int(f'1{n}1', 16)
```

```
    if bin(P).count('1') == 6:
```

```
        break
```

```
    n += 1
```

```
print(n)
```

**Ответ:**

3

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

---

## Часть 2

### Задача 1

#### Калькулятор

*Джонни работает инженером в успешной компании по производству электроники. Ранее он уже решил задачу по максимизации количества камер на телефоне. Теперь ему дали более сложную задачу, а именно разработать калькулятор, но не обычный, а особенный.*

У калькулятора есть дисплей, на котором может отображаться число. Так же у этого калькулятора должно быть несколько кнопок с числами в диапазоне от 0 до  $n-1$ .

Изначально дисплей не отображает числа. Если нажать на любую кнопку с числом, оно отобразится на дисплее.

Еще на калькуляторе есть кнопка «do magic» — прибавляет к числу на дисплее число  $k$  по модулю  $n$  и отображает полученное число. Кнопку можно нажимать несколько раз. Более формально  $(x+k \times r) \% n$ . Где  $r$  ( $r \geq 0$ ) — количество раз которые мы прибавили.  $\%$  — операция деления по модулю.

*К сожалению, в компании ограничен бюджет, поэтому Джонни нужно поместить на калькулятор минимальное количество кнопок с числами, таких чтобы с помощью нажатия этих кнопок можно было получить любое число в диапазоне от 0 до  $n-1$ .*

#### Входные данные

В первой и единственной строке даны два числа  $k$  ( $1 \leq k \leq 10^5$ ) и  $n$  ( $1 \leq n \leq 10^5$ )

#### Выходные данные

В ответе выведите минимальное количество кнопок с числами, которое нужно добавить Джонни.

#### Примеры

*Входные данные*

3 5

*Выходные данные*

1

*Входные данные*

2 8

*Выходные данные*

2

#### Пример решения:

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС

---

```
typedef long long ll;
```

```
int main() {
```

```
    ll k, n; cin >> k >> n;
```

```
    cout << __gcd(k, n) << endl;
```

```
    return 0;
```

```
}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

---

## **Задача 2**

### **Биты**

У вас есть все возможные  $n$ -разрядные целые неотрицательные числа в двоичной системе счисления и  $q$  запросов.

Порядок записи двоичного числа от старшего к младшему, то есть старший бит имеет 0 индекс, а младший  $n-1$ . Например, число 14 в десятичной системе будет представлено как 1110, а 56 как 111000

Каждый запрос имеет вид «L R». Вам нужно найти количество чисел, которые удовлетворяют условию: первые  $L$  битов должны быть нулями, и последние  $R$  битов должны быть нулями.

Так же выведите максимальное число, которое соответствует условию в десятичной системе счисления.

Так как числа могут быть очень большими — выведите их по модулю  $10^9+7$

### **Входные данные**

В первой строке даны два числа  $n$  ( $1 \leq n \leq 10^5$ ) и  $q$  ( $1 \leq q \leq 10^5$ )

В последующих  $q$  строках даны два числа  $L$  и  $R$  ( $0 \leq L, R \leq n$ )

### **Выходные данные**

В  $q$  строках выведите по 2 числа: количество чисел и максимально возможное число, удовлетворяющие условию задач.

### **Примеры**

*Входные данные*

3 1

1 1

*Выходные данные*

2 2

*Входные данные*

5 1

0 4

*Выходные данные*

2 16

*Входные данные*

5 1

0 4

*Выходные данные*

4 24

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

---

*Входные данные*

3 4  
2 2  
1 2  
2 1  
3 3

*Выходные данные*

1 0  
1 0  
1 0  
1 0

**Пример решения:**

```
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;

int main() {
    ll n, q; cin >> n >> q;
    ll mod = 1e9 + 7;
    vector<ll> cnt(n+1);
    vector<ll> pref(n+1);

    cnt[0] = 1;
    cnt[1] = 2;
    for(ll i = 2; i <= n; ++i) {
        cnt[i] = (cnt[i-1]*2)%mod;
    }
    for (ll i = 1; i <= n; ++i) {
        pref[i] = (pref[i-1] + cnt[i-1])%mod;
    }

    while(q--) {
        ll l, r; cin >> l >> r;
        if (l + r >= n) {
            cout << 1 << " " << 0 << endl;
        } else {
            ll len = n - r - l;
            cout << cnt[len] << " " << (pref[len]*cnt[r])%mod << endl;
        }
    }
}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

---

### Задача 3

#### Время

Блогер Куки готовит новый ролик о классных вещах, которые он купил в интернете. Куки приобрел необычные часы, но он забыл проверить их работу. После запуска часов оказалось, что они неисправны!

Всего на часах  $n$  делений, а большая и маленькая стрелки переключаются на  $x$  и  $y$  делений вперед в час соответственно. Стрелки мгновенно меняют свое положение, т.е. если стрелка находится в положении  $i$ , то через час будет иметь положение  $i+x$ , и стрелки не будут указывать на деления между  $i$  и  $i+x$ .

Изначально стрелки указывали в одно положение. Куки стало интересно, когда стрелки часов вновь будут указывать в одно положение.

Он не может сам решить эту задачу, потому что у него лапки. Помогите Куки решить эту задачу.

#### Входные данные

В первой и единственной строке заданы три целых числа  $n, x, y$  ( $1 \leq n, x, y \leq 1000$ ).

#### Выходные данные

Выведите через сколько часов стрелки вновь будут указывать на одно деление.

#### Примеры

*Входные данные*

24 7 3

*Выходные данные*

6

#### Пример решения:

```
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

int main() {
    ll n, x, y; cin >> n >> x >> y;
    ll xx = 0;
    ll yy = 0;
    for(ll i = 1; i <= n; ++i) {
        xx = (xx + x)%n;
        yy = (yy + y)%n;
        if (xx == yy) {
            cout << i << endl;
            return 0;
        }
    }
}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

---

```
}  
return 0;  
}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

---

## Вариант 2

### Часть 1

#### Задача 1

##### Оцифровка

Работник архива отсканировал 30 старых фотографий на цветном сканере. После этого объединил в один документ (презентацию) и сделал подписи к каждой фотографии. Общий размер документа стал равен 30996 КиБ.

Каждая страница имеет разрешение 1366 на 768 пикселей и глубину цвета 8 бит.

Определите сколько было напечатано символов, если использовалась четырехбайтная кодировка Unicode.

Объем служебной информации в презентации - 256 КиБ.

*Примечание: 1 КиБ = 1024 байт*

##### Пример решения:

```
#!/usr/bin/python3
#Разрешение отсканированных страниц
x = 1366
y = 768

#Глубина цвета
deep = 8

#Количество страниц
n = 30
#Размер документа без служебной информации
N = 30740 - 256

#Размер одного символа
encoding = 32

print((N * 1024 * 8 - x * y * deep * n) // encoding)
```

##### Ответ:

1280

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

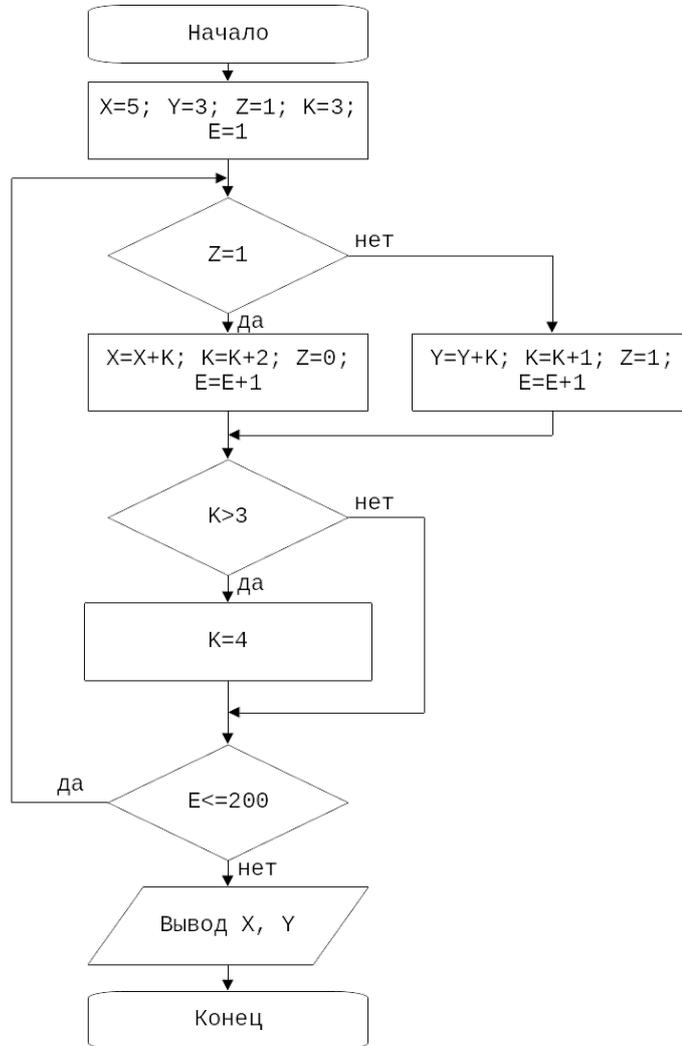
---

### Задача 2

#### Алгоритм

Дана блок-схема алгоритма.

В ответ запишите модуль разницы чисел  $X$  и  $Y$  после завершения работы алгоритма.



#### Пример решения:

```
#!/usr/bin/python3
```

```
x = 5
```

```
y = 3
```

```
z = 1
```

```
k = 3
```

```
e = 1
```

```
while e <= 200:
```

```
    if z == 1:
```

```
        x += k
```

```
        k += 2
```

```
        z = 0
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС

---

```
e += 1  
else:  
    y += k  
    k += 1  
    z = 1  
    e += 1  
if k > 3:  
    k = 4
```

```
print(abs(x - y))
```

**Ответ:**

1

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

---

### Задача 3

#### Равенство

Определите, при каком минимальном значении  $N$  сумма цифр в двоичном представлении числа  $P$  будет равна 7?

$$1N1_4 + 1N1_8 + 1N1_{16} = P.$$

*Примечание: При  $N=0$ , выражение будет иметь вид:  $1014 + 1018 + 10116 = P$*

#### Пример решения:

```
#!/usr/bin/python3
```

```
n = 0
```

```
while 1:
```

```
    P = int(f'1{n}1', 4) + int(f'1{n}1', 8) + int(f'1{n}1', 16)
```

```
    if bin(P).count('1') == 7:
```

```
        break
```

```
    n += 1
```

```
print(n)
```

**Ответ:**

1

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
9 КЛАСС**

---

**Часть 2**

Единые задания для обоих вариантов, см. Вариант 1 Часть 2.