

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

---

**Вариант 1**

**Часть 1**

**Задача 1**

**Количество единиц**

Сколько единиц в двоичной записи числа  $x$ ?

$$x = 315_6 + 224_6$$

**Пример решения:**

1.

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

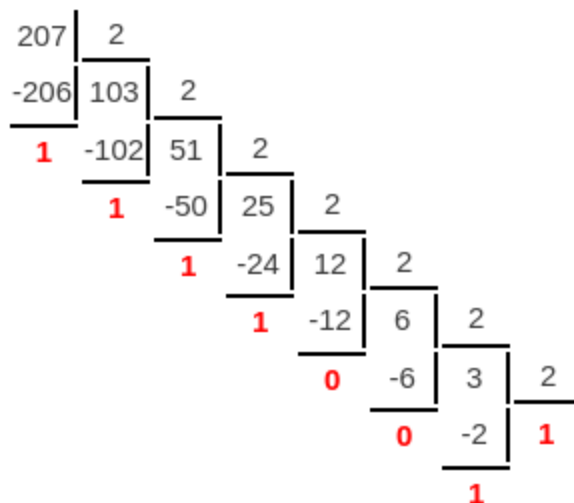
```
bin(int('315', 6) + int('224', 6)).count('1')
```

2.

$$315_6 = 3 \cdot 6^2 + 1 \cdot 6^1 + 5 \cdot 6^0 = 108 + 6 + 5 = 119_{10}$$

$$224_6 = 2 \cdot 6^2 + 2 \cdot 6^1 + 4 \cdot 6^0 = 72 + 12 + 4 = 88_{10}$$

$$119 + 88 = 207$$



$$207_{10} = 11001111_2$$

**Ответ:**

6

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

---

## Задача 2

### Кодирование

Ниже представлены 10 числовых строк. Необходимо получить 10 строк посредством RLE кодирования (Кодирование длинных серий). В ответ запишите разницу в длине между самой длинной и самой короткой строкой, полученными после кодирования.

*Пример кодирования длинных серий:*

WWWWWWWWBWWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWW  
WWWWW → 9W3B24W1B14W

222322333124

432311233344

144412131131

321233422414

323443443221

131241244234

212144213211

114343132224

43333442221

323323423413

### Пример решения:

1.

```
#!/usr/bin/python3
data = """222322333124
432311233344
144412131131
321233422414
323443443221
131241244234
212144213211
114343132224
43333442221
323323423413""".split('\n')
```

```
out = []
```

```
for line in data:
    prev_letter = ""
    encode = ""
    c = 1
    for letter in line:
        if letter != prev_letter:
            if prev_letter:
                encode += str(c) + prev_letter
```

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

---

```
c = 1
prev_letter = letter
else:
    c += 1
    encode += str(c) + prev_letter

out.append(encode)

sorted_out = sorted(out, key=lambda x: len(x))

print(len(sorted_out[-1]) - len(sorted_out[0]))
```

2.

Также решается вручную, на внимательность.

**Ответ:**

12

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

---

### Задача 3

#### Микробиология

Петя и Вася изучали микробиологию. Спустя некоторое время они утомились и решили поиграть в слова. Их задача состояла в том, чтобы составить максимально возможное количество комбинаций из 6-ти букв, входящих в слово "Микробиология" так, чтобы все буквы в комбинации были различны и все комбинации отличались друг от друга хотя бы одной буквой. Петя выиграл эту игру, записав все возможные комбинации. Сколько у него их получилось?

Комбинации, состоящие из одинаковых букв, но стоящих в разном порядке, считаются различными.

#### Пример решения:

1.

```
#!/usr/bin/python3
from math import factorial

s = 'Микробиология'

letter_count = len(set(s.lower()))

print(factorial(letter_count) // factorial(6))
```

2.

В слове Микробиология 9 различных букв, Ответ - число размещений из 9 по 6.  
 $A(n,m) = n! / (n-m)! = 9! / (9-6)! = 9!/3! = 60480$

#### Ответ:

60480

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

---

## Часть 2

### Задача 1

#### Стрелки

БиМО закинул Финна и Джейка внутрь себя, чтобы поиграть в видеоигру под названием «Хранители солнечного света», где сражаются с трёхмерными цифровыми врагами.

У Финна и Джейка есть  $N$  стрелок.

Каждая из них направлена в одну из четырех возможных сторон (L, U, R, D), где L — влево, U — вверх, R — вправо, а D — вниз.

За один ход они могут повернуть одну из стрелок на 90 градусов вправо или влево.

Так при повороте стрелок (L, U, R, D) вправо, получится (U, R, D, L).

А при повороте стрелок (L, U, R, D) влево, получится (D, L, U, R).

Только комбо-удар может убить всех врагов. Комбо-ударом называется комбинация стрелок, которые направлены в одну сторону.

Помогите определить за какое минимальное количество ходов можно направить все стрелки в одну сторону, чтобы совершить комбо-удар.

#### Входные данные

В первой строке дано целое число  $N(1 \leq N \leq 10^5)$  — количество стрелок

Во второй строке дана строка  $s$  размера  $N$ , состоящая из символов (L, U, R, D)

#### Выходные данные

В единственной строке выведите число — минимальное количество ходов, за которое все стрелки будут смотреть в одну сторону.

#### Примеры

*Входные данные*

5

RRRRR

*Выходные данные*

0

*Входные данные*

5

RURRD

*Выходные данные*

2

#### Пример решения:

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

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

---

```
typedef long long ll;

using namespace std;

unordered_map<char, int> charToInt = {{'L', 0}, {'U', 1}, {'R', 2}, {'D', 3}};

int dist(const string& s, char ch) {
    int res = 0;
    for(int i = 0; i < s.size(); ++i) {
        if (s[i] != ch) {
            if (charToInt[s[i]]%2 == charToInt[ch]%2) {
                res += 2;
            } else {
                res++;
            }
        }
    }
    return res;
}

int main() {
    int n; cin >> n;
    string s; cin >> s;

    cout << min({dist(s, 'L'), dist(s, 'U'), dist(s, 'R'), dist(s, 'D')});

    return 0;
}
```

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

---

## Задача 2

### Пересечение прямоугольников

Даня и Егор спорят о том, что лучше симпл-димпл или попит.

Давайте представим, что мнения ребят это прямоугольники на координатной плоскости, а стороны прямоугольников расположены параллельно осям координат. Ребята хотят посчитать насколько их мнения совпадают. Совпадением мнений будем называть площадь пересечения этих прямоугольников.

Ваша задача посчитать величину совпадения мнений.

### Входные данные

В первой строке даны 4 целых числа  $x_1, y_1, x_2, y_2$  ( $-1000 \leq x_1, y_1, x_2, y_2 \leq 1000$ ), где  $x_1, y_1$  — это координаты левого верхнего угла первого прямоугольника, а  $x_2, y_2$  — координаты правого нижнего угла первого прямоугольника.

Во второй строке даны 4 целых числа  $x_3, y_3, x_4, y_4$  ( $-1000 \leq x_3, y_3, x_4, y_4 \leq 1000$ ), где  $x_3, y_3$  — это координаты левого верхнего угла второго прямоугольника, а  $x_4, y_4$  — координаты правого нижнего угла второго прямоугольника.

### Выходные данные

Выведите одно число — площадь пересечения прямоугольников.

### Примеры

*Входные данные*

0 10 10 0

0 20 20 0

*Выходные данные*

100

*Входные данные*

-1 1 0 0

0 0 2 -1

*Выходные данные*

0

### Пример решения:

```
#include <iostream>
#include <vector>
#include <map>
#include <algorithm>
#include <math.h>
#include <set>
using namespace std;
```

```
//#define _DEBUG
```

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

---

```
struct Point {
    int x;
    int y;
};

struct Rect {
    Point right_up;
    Point left_up;
    Point right_down;
    Point left_down;
    Rect() {}
    Rect(Point p1, Point p2) {
        Point p3 = {p1.x, p2.y};
        Point p4 = {p2.x, p1.y};
        vector<Point> v = {p1,p2,p3,p4};
        for(int i = 0; i < v.size(); i++) {
            for(int j = i+1; j < v.size(); j++) {

                if (v[i].y >= v[j].y) swap(v[i],v[j]);
                if (v[i].y == v[j].y && v[i].x >= v[j].x) swap(v[i],v[j]);

            }
        }
        // for(auto i : v) {
        //     cout << i.x << ' ' << i.y << endl;
        // }
        //cout << endl;
        left_down = v[0];
        right_down = v[1];
        left_up = v[2];
        right_up = v[3];
    }
};

void solve() {
    Point p1,p2; cin >> p1.x >> p1.y >> p2.x >> p2.y;
    Point p3,p4; cin >> p3.x >> p3.y >> p4.x >> p4.y;
    Rect rect1(p1,p2);
    Rect rect2(p3,p4);

    int left = max(rect1.left_down.x, rect2.left_down.x);
    int top = min(rect1.right_up.y, rect2.right_up.y);
    int right = min(rect1.right_up.x, rect2.right_up.x);
    int bottom = max(rect1.left_down.y, rect2.left_down.y);
    //cout << left << ' ' << top << ' ' << right << ' ' << bottom << endl;
    int width = right - left;
    int height = top - bottom;
```



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

---

```
if (width < 0 || height < 0)
{
    cout << 0 << endl;
    return;
}

cout << width * height << endl;

}

int main() {
    ios_base::sync_with_stdio(0); cin.tie(0);cout.tie(0);
    #ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif
    solve();
    return 0;
}
```

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

---

### **Задача 3**

#### **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  $p-1$ .

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  $p$  и отображает полученное число. Кнопку можно нажимать несколько раз. Более формально  $(x+k \times r) \% p$ . 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  $p-1$ .

#### **Входные данные**

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

#### **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;
```

```
typedef long long ll;
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
ИНФОРМАТИКА. ОТБОРОЧНЫЙ ТУР  
8 КЛАСС

---

```
int main() {  
  
    ll k, n; cin >> k >> n;  
  
    cout << __gcd(k, n) << endl;  
  
    return 0;  
}
```





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

---

```
if letter != prev_letter:
    if prev_letter:
        encode += str(c) + prev_letter
    c = 1
    prev_letter = letter
else:
    c += 1
encode += str(c) + prev_letter

out.append(encode)

sorted_out = sorted(out, key=lambda x: len(x))

print(len(sorted_out[-1]) - len(sorted_out[0]))
```

2.

Также решается вручную, на внимательность.

**Ответ:**

10

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

---

### Задача 3

#### Микроэлектроника

Петя и Вася изучали микроэлектронику. Спустя некоторое время они утомились и решили поиграть в слова. Их задача состояла в том, чтобы составить максимально возможное количество комбинаций из 6-ти букв, входящих в слово "Микроэлектроника" так, чтобы все буквы в комбинации были различны и все комбинации отличались друг от друга хотя бы одной буквой. Петя выиграл эту игру, записав все возможные комбинации. Сколько у него их получилось?

#### Пример решения:

1.

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

```
from math import factorial
```

```
s = 'Микроэлектроника'
```

```
letter_count = len(set(s.lower()))
```

```
print(factorial(letter_count) // factorial(6))
```

2.

В слове Микроэлектроника 11 различных букв, Ответ - число размещений из 11 по 6.

$$A(n,m) = n! / (n-m)! = 11! / (11-6)! = 11!/5! = 332640$$

#### Ответ:

332640

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

---

**Часть 2**

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