

**Заключительный этап Олимпиады школьников «Шаг в будущее» по профилю «Инженерное дело» специализации «Техника и технологии (программирование)»
(общеобразовательный предмет информатика), весна 2021 год**

**11 класс
Вариант 2**

Задача 1 (5 баллов)

Условие

Для заданного целого числа от 1 до 10^8 определить, какое количество входящих в него цифр одновременно являются его делителями.

Входные данные: одно целое положительное число, не превышающее 10^8 .

Выходные данные: одно целое число - количество цифр, из которых состоит исходное число и на которые его можно разделить.

Пример

Входные данные	Результат
204	2
5555	4

Проверочные тесты

Входные данные	Ожидаемый результат
204	2
5555	4
1	1
200	1
23	0

Пример решения

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

int main(){
unsigned long num=0, a=0, n=0;
cin>>num;
a=num;
do{
    if(a%10!=0) if(num%(a%10)==0) n++;
    a/=10;
}while(a>0);
cout<<n<<endl;
return 0;
```

}

Задача 2 (8 баллов)

Условие

16-разрядное число преобразовали путём циклического сдвига вправо на N разрядов.

Известны исходное число и число после сдвига, записанные в 16-ричной системе счисления. К сожалению, при выполнении операции произошёл сбой, в результате которого один из разрядов был искажён. Требуется определить, в каком разряде нового числа допущена ошибка, и указать его номер и правильное значение.

Разряды нумеруются справа налево (от младшего к старшему) от 1 до 16.

Входные данные: два 16-разрядных числа, записанных в 16-ричной системе счисления и состоящих ровно из 4-х цифр 0..F, и количество разрядов, на которые осуществлён сдвиг, от 1 до 15. Цифры A..F всегда задаются заглавными буквами. Исходные значения записаны в одну строку через пробел.

Выходные данные: номер искажённого разряда во втором числе и его верное значение.

Пример

Входные данные	Результат
7777 77F7 12	8 0
AAAA A8AA 6	10 1

Проверочные тесты

Входные данные	Ожидаемый результат
7777 77F7 12	8 0
AAAA A8AA 6	10 1
1111 1110 4	1 1
1000 3000 15	13 0
AAAA 5545 1	5 1

Пример решения

```
#include <iostream>
#include <string>
#include "math.h"
using namespace std;

typedef long long ll;

ll CharToLL(char c) {
    ll v = c - '0';
    if(v > 9) {
        return c - 'A' + 10;
    }
    return v;
}
```

```

}

ll StringToLL(string s) {
    ll v = 0;
    ll p = 1;
    for(int i = s.length() - 1; i >= 0; i--) {
        char c = s[i];
        v += p * CharToLL(c);
        p *= 16;
    }
    return v;
}

int main() {
    string s1, s2;
    ll n;
    cin >> s1 >> s2 >> n;
    ll v1 = StringToLL(s1);
    ll v2 = StringToLL(s2);
    ll vReal = v1;
    for(ll i = 0; i < n; i++) {
        ll last = vReal & 0b1;
        last = last << 15;
        vReal = last | (vReal >> 1);
    }
    for(ll i = 0; i < 16; i++) {
        ll lastReal = vReal & 0b1;
        ll last2 = v2 & 0b1;
        if(last2 != lastReal) {
            cout << i + 1 << " " << lastReal << "\n";
            break;
        }
        vReal = vReal >> 1;
        v2 = v2 >> 1;
    }
    return 0;
}

```

Задача 3 (10 баллов)

Условие

Задана строка, состоящая из цифр и знаков "+" и "-". Требуется определить, можно ли расставить в ней скобки так, чтобы получившееся выражение было равно нулю.

Входные данные: строка символов, длина которой не превышает 20.

Выходные данные: выражение, равное нулю, если его возможно получить, и слово NO, если невозможно.

Пример

Входные данные	Результат
3-1+2	3-(1+2)
4+3+2	NO

Проверочные тесты

Входные данные	Ожидаемый результат
3-1+2	3-(1+2)
4+3+2	NO
1-1	1-1
4-2+3+1	4-(2+3)+1
9-7+2-2+3+5	9-(7+2)-(2+3)+5

Пример решения

```
def check(line):
    if eval(line) == 0:
        return True
    return False

def func(text_):
    if check(text_):
        return text_
    plus_count = text_.count("+")
    minus_count = text_.count("-")

    indx_start = 0
    for _ in range(minus_count):
        text1 = text_
        indx = text1.index('-', indx_start)
        text1 = f"{text1[:indx + 1]}({text1[indx + 1:]}"

        count = text1.count("+", indx) + text1.count("-", indx + 1)
        for i in range(count):
            text = text1
            if "+" in text[indx + 1:] and "-" in text[indx + 1:]:
                indx2 = min(text.index("+", indx + 1), text.index("-", indx +
1))

            elif "+" in text[indx + 1:]:
                indx2 = text.index("+", indx + 1)
            else:
                indx2 = text.index("-", indx + 1)

            if i + 1 == count:
                indx2 = -1
            else:
                print(text[indx2 + 1:], i, count)
                if "+" in text[indx2 + 1:] and "-" in text[indx2 + 1:]:
                    indx2 = min(text.index("+", indx2 + 1), text.index("-",
indx2 + 1))

                elif "+" in text[indx2 + 1:]:
                    indx2 = text.index("+", indx2 + 1)
                else:
                    indx2 = text.index("-", indx2 + 1)
            if indx2 == -1:
                text = f"{text})"
```

```

        else:
            text = f"{text[:indx2]}{text[indx2:]}"
            if check(text):
                return text
    else:
        return "NO"

text_input = input()
print(func(text_input))

```

Задача 4 (12 баллов)

Условие

Для заданной прямоугольной площадки $N \times M$ требуется определить, можно ли всю её площадь плотно заполнить блоками размера $P \times Q$ так, чтобы они не перекрывались и их не приходилось обрезать.

Блоки прямоугольные, допускается их поворачивать на 90 градусов.

Входные данные: 4 натуральных числа N , M (каждое не превышает 50), P , Q (не превышают 5), записанные через пробел.

Выходные данные: необходимое количество блоков, если ими можно заполнить площадку, или 0, если заполнить площадку с соблюдением указанных ограничений невозможно.

Пример

Входные данные	Результат
5 4 1 2	10
3 3 2 2	0

Проверочные тесты

Входные данные	Ожидаемый результат
2 4 1 3	0
4 3 2 1	6
1 1 1 1	1
4 7 3 3	0
12 7 4 3	7

Пример решения

```

#include <iostream>
using namespace std;

typedef long long ll;

bool fdiv(ll a, ll b) {
    return a % b == 0;
}

```

```

}

int main() {
    ll M, N, P, Q;
    cin >> M >> N >> P >> Q;
    if((fdiv(M, P) && fdiv(N, Q))
        || fdiv(M, Q) && fdiv(N, P)) {
        cout << (M*N) / (P*Q);
    } else {
        cout << 0;
    }
    return 0;
}

```

Задача 5 (15 баллов)

Условие

Пятиклассник Вася отправился на прогулку, взяв с собой сотовый телефон. Перед этим его родители для Васиной безопасности и собственного спокойствия нашли и установили специальную программу, которая передаёт им сведения о ближайших к Васе антеннах сотовой связи.

Помогите Васиным родителям написать программу, которая по истории Васиных перемещений определит, какое расстояние прошёл Вася.

Входные данные: в первой строке записано целое число N (от 3 до 10) - количество вышек сотовой связи на Васином маршруте. Далее идёт N строк, в которых через пробел записаны координаты X и Y соответствующей антенны (в метрах, целые, от -1000 до 1000 каждая).

В следующей строке записано целое число K (от 2 до 100) - количество точек Васиной прогулки, в которых передавались сведения о его местоположении. Далее идёт K строк, в которых через пробел указаны 7 чисел: время в минутах с начала прогулки (целое от 0 до 300), номера и мощности сигнала с трёх ближайших к Васе антенн. Антенны нумеруются по порядку ввода с 1, мощность задаётся в виде процентов от 1 до 100, целым числом. Первая точка имеет время 0, последняя соответствует концу прогулки.

В рамках задачи считать, что:

- Вася гуляет по ровной местности;
- между зафиксированными точками идёт только по прямой;
- все антенны находятся на уровне земли, имеют одинаковую мощность и Васин телефон теряет с ними связь на расстоянии в 1000 метров.

Выходные данные: расстояние, пройденное Васей за прогулку, округлённое со сотен метров.

Пример

Входные данные	Результат
4 0 0 500 0 0 1000 500 1000 3	1300

0 1 90 3 10 2 49 10 1 10 3 90 4 49 30 3 49 4 90 2 10	
--	--

Проверочные тесты

Входные данные	Ожидаемый результат
4 0 0 500 0 0 1000 500 1000 3 0 1 90 3 10 2 49 10 1 10 3 90 4 49 30 3 49 4 90 2 10	1300
3 100 0 200 100 300 0 2 0 1 90 2 90 3 78 5 1 78 2 90 3 90	200
3 100 0 200 100 300 0 2 0 1 90 2 90 3 78 5 1 78 2 78 3 90	300
7 100 100 200 100 300 100 100 500 200 500 300 500 100 900 3 0 1 86 2 78 3 68 10 7 86 5 46 6 49 30 3 90 2 86 3 78	2000

7	1400
100 100	
200 100	
300 100	
100 500	
200 500	
300 500	
100 900	
5	
0 1 86 2 78 3 68	
5 1 80 4 80 2 78	
10 6 80 3 80 5 78	
15 6 90 5 86 4 78	
25 3 90 2 86 1 78	

Пример решения

```

from math import sqrt
N = int(input())
cells = []
for i in range(N):
    X, Y = map(int, input().split())
    cells.append([X, Y])

K = int(input())
data_dict = {}
for i in range(K):
    inp = list(map(int, input().split()))

    data_dict[inp[0]] = [[inp[j], inp[j+1]] for j in range(1, len(inp), 2)]

db = []
for key in sorted(data_dict.keys()):
    db.append(data_dict[key])

points = []
for i in range(K):
    data = db[i]
    y1b = (100-data[0][1])/100*1000
    y2b = (100-data[1][1])/100*1000
    y3b = (100-data[2][1])/100*1000
    i1 = data[0][0]
    i2 = data[1][0]
    i3 = data[2][0]
    x1 = cells[i1-1][0]
    x2 = cells[i2-1][0]
    x3 = cells[i3-1][0]
    y1 = cells[i1-1][1]
    y2 = cells[i2-1][1]
    y3 = cells[i3-1][1]

    if x2-x1 == 0:
        a = (x3*x3-x2*x2) + (y3*y3-y2*y2) + y2b*y2b - y3b*y3b
        b = 2*(x3-x2)
        z = 2*(y3-y1)-4*(y3-y2)*(x3-x1)/b
    else:

```

```
a = (x2*x2-x1*x1) + (y2*y2-y1*y1) + y1b*y1b - y2b*y2b
b = 2*(x2-x1)
z = 2*(y3-y1)-4*(y2-y1)*(x3-x1)/b
```

```
c = -(x3-x1)*2*a/b+ y3*y3 - y1*y1 + x3*x3 -x1*x1 + y1b*y1b -y3b*y3b
```

```
y = c/z
```

```
if x2-x1 == 0:
    x = (a-2*y*(y3-y2))/b
else:
    x = (a-2*y*(y2-y1))/b
```

```
points.append([x, y])
```

```
answer = 0
for i in range(len(points)-1):
    x1 = points[i][0]
    y1 = points[i][1]
    x2 = points[i+1][0]
    y2 = points[i+1][1]
    answer += sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1))

print(int(round(answer, -2)))
```