

## Задача 1

Петя и Вася строят каждый свою башню из кубиков одинакового размера, устанавливая один кубик на другой. Две их башни стоят рядом, поэтому при построении башен друзья мешают друг другу. К некоторому моменту времени Петя построил башню высотой  $X$  кубиков, а Вася – высотой  $Y$  кубиков ( $X > Y$ ). Чтобы дальше не мешать друг другу, ребята договорились строить башни по очереди. Одну минуту – Вася, одну минуту – Петя и так далее. Начинает строить Вася. Он за минуту укладывает  $A$  кубиков, а Петя –  $B$  кубиков ( $A > B$ ). Через сколько минут башня Пети перестанет быть выше башни Васи?

### Входные данные

Четыре целых числа, записанные через пробел  $X, Y, A, B$  ( $1 \leq X, Y, A, B \leq 10^9$ )

### Выходные данные

Целое число – количество минут, через которое башня Пети станет не выше башни Васи.

### Пример

Входные данные	Выходные данные
10 4 3 1	5

### Решение

```
x, y, dy, dx = map(int, input().split())
k = 0
d = dy - dx
if y + dy >= x:
    print(1)
else:
    print ((x - y) - dy + d - 1) // d * 2 + 1)
```

## Задача 2

Игра «Космическое домино».

Правила.

1. В игре участвуют только **ТРЕХЗНАЧНЫЕ** числа.
2. Перед началом игры для каждого игрока случайным образом генерируется некоторый диапазон трехзначных чисел, из которого и **ТОЛЬКО** из него он может выбирать числа для продолжения игры.
3. Первые два стартовых числа генерируются тоже компьютером.
4. Игроки ходят по очереди, доставляя сопряженное число из своего диапазона (если такое есть) к левому или правому концу цепочки.
5. Если у игрока нет в диапазоне числа, сопрягаемого ни с одним из концов цепочки, то игра завершается его проигрышем.

Определение: число М называется **сопрягаемым** с числом N, если оно построено по следующим правилам:

1. Если N нечетное, то число М начинается с нечетной цифры, если N – четное, то с четной (но не с 0!)
2. Последние две цифры числа М есть сумма цифр числа N.

Заметим, что оба числа являются трехзначными!

Например, если  $N = 213$ , тогда М может быть 106, 306, 506, 706 или 906. Для числа  $N = 914$ , число М может быть только 214, 414, 614 или 814.

Примечание: числа в цепочке могут повторяться.

### Входные данные:

L и R два натуральных трехзначных числа через пробел в одной строке. Левое и правое число в цепочке соответственно.

A и B два натуральных трехзначных числа через пробел в одной строке. Диапазон, который выпал игроку в начале игры.  $A < B$ . Число A или B также может быть выбрано игроком для хода, если является сопрягаемым с концом цепочки.

### Выходные данные:

На первой строчке количество чисел, сопрягаемых с концами цепочки из диапазона игрока.

На второй строке минимальное из этих чисел с указанием перед ним без пробела литеры “L”, если его надо поставить к левому концу, и литеры “R”, если к правому. Если число можно добавить в любой конец, то ставим его в **левый** конец.

Входные данные	Вывод	Примечание
213 914 306 520	3 L306	На данный момент игровая цепочка выглядит: 213 ... 914 Множество спрягаемых с концами цепочки чисел: 106, 306, 506, 706, 906, 214, 414, 614 или 814. Из них в распоряжении игрока только 3 числа: 306, 506, 414. Наименьшее из них 306, и оно сопрягается с левым концом. После хода цепочка будет иметь вид: 306 213 ... 914
213 914 350 400	0	Множество спрягаемых с концами цепочки чисел: 106, 306, 506, 706, 906, 214, 414, 614 или 814. К сожалению в диапазон игрока не входит ни одно из чисел, поэтому он проиграл, так как не может сделать ход.

914 806 600 800	1 L614	Множество спрягаемых с концами цепочки чисел совпало: 214, 414, 614 или 814. В диапазон игрока попало только одно число 614, его можно поставить и слева, и справа. Согласно стратегии, описанной выше, ставим его слева.
--------------------	-----------	--

### Решение

```
def linking(n): # функция получения всех чисел, сопрягаемых с n
    s = sum([int(x) for x in str(n)])
    if n % 2 == 0:
        m = [ y * 100 + s for y in [2,4,6,8]]
    else:
        m = [ y * 100 + s for y in [1,3,5,7,9]]
    return m

left, right = map(int, input().split())
a, b = map(int, input().split())
mL = linking(left)
mR = linking(right)
res = [x for x in mL + mR if a <= x <= b] # выбираем из сопрягаемых с концами чисел те,
                                         # которые есть в диапазоне у игрока
print(len(set(res))) # убираем повторы с помощью set и считаем количество альтернативных ходов
if res:
    mn = min(res)
    if mn in mL:
        print("L"+str(mn))
    elif mn in mR:
        print("R"+str(mn))
```

### Задача 3

Исполнитель Говорун действует в трехмерном пространстве с декартовой системой координат. Исполнитель Говорун умеет исполнять только одну команду **Перелететь на (a,b,c)** (где ***a, b, c*** - целые числа), перемещающую исполнителя Говоруна из точки с координатами (***x, y, z***) в точку с координатами (***x+a, y+b, z+c***). Если числа ***a, b, c*** положительные, то значения соответствующей координаты увеличивается, если отрицательные - уменьшается. Исполнитель Говорун был запущен с корабля "Синяя чайка" (координаты корабля - целые числа) по определенному маршруту с сигналом ***SOS***. Исполнитель не может удаляться от корабля "Синяя чайка" на расстояние более, чем ***R***.

Определите сколько различных точек с целочисленными координатами принадлежит траектории Говоруна до завершения его миссии, считая начальную и конечную точки. Миссия исполнителя завершается, если:

- выполнены все команды;

- исполнитель нашел корабль "Пегас" (то есть точка с координатами корабля принадлежит траектории исполнителя);

- исполнитель удалился от корабля "Синяя чайка" на расстояние большее, чем  $R$ .

### **Входные данные:**

1-я строка содержит координаты корабля "Синяя чайка" и расстояние  $R$  (все числа целые и по модулю не более  $10^6$ ,  $R$  - натуральное);

2-я строка содержит координаты корабля "Пегас"

3-я строка содержит число  $N$  - количество команд ( $N \leq 10^5$ ).

В следующих  $N$  строках записаны параметры для команд исполнителя - три целых числа.

### **Выходные данные**

Ответ на задачу - количество различных целочисленных точек, в которых успеет побывать Исполнитель до завершения программы

### **Примеры:**

входные данные	Выходные данные	Пояснение
0 0 0 400 0 0 100 3 3 3 3 2 -3 -3 -6 0 0	10	Миссия Говоруна завершилась после выполнения всех команд. Говорун посетил следующие "целочисленные" точки (в порядке посещения): (0,0,0); (1,1,1); (2,2,2); (3,3,3); (5,0,0); (4,0,0); (3,0,0); (2,0,0); (1,0,0); (0,0,0); (-1,0,0) Различных будет 10
0 0 0 4 0 0 100 3 3 3 3 2 -3 -3 -6 0 0	3	Миссия Говоруна завершилась при выполнении 1 команды из-за удаления на расстояние более чем 4. Говорун посетил следующие "целочисленные" точки (в порядке посещения): (0,0,0); (1,1,1); (2,2,2). Точка (3, 3, 3) удалена от (0, 0, 0) на расстояние более, чем 4
0 0 0 400 4 0 0 3 3 3 3 2 -3 -3 -6 0 0	6	Миссия Говоруна завершилась после прибытия на корабль "Пегас". Говорун посетил следующие "целочисленные" точки (в порядке посещения): (0,0,0); (1,1,1); (2,2,2); (3,3,3); (5,0,0); (4,0,0) - прибытие на "Пегас" Различных будет 6

```
def nod(a,b):  
    while b>0 :  
        a,b=b,a%b
```

```

        return a

x,y,z,R=map(int,input().split())
xx,yy,zz=map(int,input().split())
R2=R*R
tt=set()
tt.add((x,y,z))
S=(x,y,z)
F=(xx,yy,zz)
N=int(input())
fl=1
for _ in range(N):
    if fl !=1 : break
    a,b,c=map(int,input().split())
    dd=nod(abs(a),abs(b))
    d=nod(dd,abs(c))
    a1,b1,c1=a//d, b//d, c//d
    for i in range(d):
        x,y,z=x+a1,y+b1,z+c1
        r=(x-S[0])**2+(y-S[1])**2+(z-S[2])**2
        if r>R2:
            fl=-1
            break
        tt.add((x,y,z))
        if (x,y,z)==F :
            fl=0
            break
    if fl !=1 : break
answer=len(tt)
print(answer)

```

#### Задача 4

Младшеклассники в процессе изучения арифметики играют в игру. Из стопки по одной тянут карточки с цифрами и выкладывают их в цепочку. Вытянутую карточку можно добавить или в начало, или в конец цепочки. Процедура повторяется, пока в стопке есть карточки. Какое наибольшее число можно собрать таким образом, если известна последовательность карточек?

#### Формат ввода

На вход программе в первой строке подается натуральное число  $N$  ( $N \leq 10000$ ) – количество карточек. Далее в  $N$  строках подаётся по одному целому числу – от 0 до 9. Гарантируется, что в первой строчке подаётся число больше 0.

#### Формат вывода

Вывести одно целое число – наибольшее возможное, которое можно получить по правилам, описанным в условии задачи.

### Пример

Ввод	Вывод
5 8 2 3 5 9	98235

### Решение

```
def count(x):  
    s = 0  
    while x>0:  
        if x%12 in [0,3,6,9]:  
            s+=1  
        x=x//12  
    return s
```

```
n = int(input())  
answer = 0  
for i in range(n):  
    x = int(input())  
    answer+=count(x)
```

```
print(answer)
```

### Задача 5

Старшеклассник Миша собирает робота, который должен ездить по лабиринту. Всего робот умеет выполнять 12 различных команд, но для нас представляют интерес четыре из них – «на клетку вперёд», «на клетку назад», «на клетку влево», «на клетку вправо», обозначенные кодами 0, 1, 2, 3 соответственно. Миша решил передавать роботу инструкции в виде цифр числа: робот получает число, переводит его в двенадцатеричную систему и выполняет соответствующие цифрам команды.

На вход подаётся  $N$  чисел с наборами команд. Сколько раз робот изменит положение, если считать, что он не встречается препятствий?

#### Формат ввода

На вход программе в первой строке подается натуральное число  $N$  ( $N \leq 10000$ ) – количество наборов команд. Далее в  $N$  строках на вход подаётся по одному целому числу в диапазоне от 0 до  $4 \cdot 10^9$  – набор двенадцатеричных команд, записанных в десятичной системе счисления.

#### Формат вывода

Вывести одно целое число – сколько раз робот изменит положение.

### Пример

Ввод	Вывод
4 144 149 140 12	7

### Решение

```
def count(x):  
    s = 0  
    while x>0:  
        if x%12 in [0,3,6,9]:  
            s+=1  
        x=x//12  
    return s
```

```
n = int(input())  
answer = 0  
for i in range(n):  
    x = int(input())  
    answer+=count(x)
```

```
print(answer)
```

### Задача 6

Корпорация RosOil организует охрану нефтепровода. Регион, в котором находится нефтепровод, слабо заселён и отличается неразвитой инфраструктурой, и поэтому Корпорация предпочитает держать в регионе небольшие группы специалистов для решения кризисных ситуаций, а рутинное наблюдение за участками нефтепровода вести с помощью летающих дронов «Кактус». Эти дроны отличаются высокой автономностью, но их запускают из одной стартовой точки, поэтому цена наблюдения за участком рассчитывается как произведение сложности участка на его порядковый номер. Нумерация начинается с единицы. На одном из участков нефтепровода нужно разместить узел дронов. На расстоянии не более  $M$  участков от узла цена наблюдения будет равна только сложности участка. Определите номер участка, на котором RosOil будет выгоднее всего разместить узел дронов.

### Формат ввода

На вход программе в первой строке подаётся натуральное число  $N$  ( $N \leq 1000000$ ) – количество участков. Во второй строке подаётся натуральное число  $M$  ( $M \leq 1000$ ) – расстояние, на котором цена наблюдения будет равна только сложности участка. Далее

в  $N$  строках подаётся по одному натуральному числу  $P_i$  ( $P_i \leq 100$ ) – сложность участка номер  $i$ .

### Формат вывода

Вывести одно натуральное число – номер участка, на котором надо разместить узел дронов. Если подходящих участков несколько, то выбрать участок с наименьшим номером.

### Пример

Ввод	Вывод
5 1 1 5 3 4 3	4

$$3+4+3+5*2+1*1=23$$

$$3*5+4+3+5+2*1+1*1=$$

### Решение

```
n = int(input())
m = int(input())
data = n*[0]
for i in range(n):
    data[i] = int(input())
mnm = sum(data)*n
pref=[0]
pref2=[0]
i = 1
for x in data:
    pref.append(pref[-1]+x*i)
    pref2.append(pref2[-1]+x)
    i+=1

for i in range(1,n+1):
    s=0
    s+=pref[max(i-m-1,0)]-pref[0]
    s+=pref2[min(i+m,n)]-pref2[max(i-m-1,0)]
    s+=pref[n]-pref[min(n,i+m)]
    if s<mnm:
        answer,mnm=i,s

print(answer)
```



