

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

Шпунтик очень любил мастерить разные вещи. Однажды он задумал соорудить пылесос для уборки помещения. Он трудился целый день и целую ночь, и только наутро пылесос был готов. Шпунтик запрограммировал пылесос на выполнение ряда команд и присвоил ему громкое имя "Винтик-1".

Исполнитель Винтик-1 умеет выполнять только две команды:

Forward - вызывает перемещение и исполнителя на 1 единицу расстояния и уменьшает заряд исполнителя на 2 единицы;

Left - вызывает изменение направления движения на 90 градусов влево и уменьшает заряд исполнителя на 1 единицу.

Испытание своего изобретения запланировал в большом зале, разлинованном на единичные клетки. В узел каждой клетки, кроме начальной, Шпунтик положил по 1 винтику. Пылесос разместил в центре вдоль одной из линий.

Определите, сколько винтиков собрал бы исполнитель Винтик-1, если известны: начальный заряд исполнителя и список команд.

Считается, что команда может быть выполнена, если размера заряда хватает на её выполнение. Гарантируется, что размер помещения достаточен для выполнения всех команд.

Входные данные:

В первой строке указан начальный заряд исполнителя и количество команд N (числа натуральные и не более 1000000). В следующих N строках записаны команды для исполнителя.

Выходные данные

Ответ на задачу - количество винтиков, которые успеет собрать исполнитель до окончания заряда или завершения программы

Примеры:

входные данные	Выходные данные	Пояснение
4 5 Left Left Left Forward Forward	0	После выполнения трех команд у исполнителя останется 1 единица заряда, которой не хватит на выполнение 4 команды. Исполнитель остался на месте.
6 4 Forward Forward Left Forward	2	После выполнения трех команд у исполнителя останется 1 единица заряда, которой не хватит на выполнение 4 команды
1000 12 Forward Left Forward Left Forward Left Forward Left Forward Left Forward Left	3	Исполнитель двигается "по кругу", поэтому побывает всего на 4 клетках, одна из которых "стартовая" (на старте винтика не было)

Решение

```
def stz(nn):
```

```

return int((nn-1)**0.5)+1

z,m=map(int,input().split())
Sdvig=[(1,0),(0,1),(-1,0),(0,-1)]
tt=set()
x,y,curs=0,0,0
tt.add((x,y))
for _ in range(m):
    ss=input()
    if 'Left' in ss and z>0:
        curs=(curs+1)%4
        z-=1
    elif 'Right' in ss and z>0:
        curs=(curs+3)%4
        z-=1
    elif 'Forward' in ss :
        tt.add((x,y))
        nn=int(ss.split()[1])
        n=stz(nn)
        if z>=n :
            z-=n
            a,b=Sdvig[curs]
            for i in range(nn):
                x+=a
                y+=b
                tt.add((x,y))
answer=len(tt)-1
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

```

n = int(input())
answer = str(input())

for i in range(1,n):
    x = int(input())
    y = int(answer[0])
    if x>=y:
        answer = str(x)+answer
    else:
        answer = answer+str(x)

print(int(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 встречает 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	7

144	
149	
140	
12	

Решение

```
def count(x):
    s = 0
    while x>0:
        if x%12 in [0,1,2,3]:
            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

Российская госкорпорация занимается реорганизацией добычи полезных ископаемых в Западной Африке. Персонал, оборудование и расходные материалы доставляются на место самолётами из нескольких аэропортов. В полночь по местному времени профессор Львов отправил запрос на срочную поставку очень ценного тяжёлого самоходного геологического оборудования. Профессор попросил аспиранта Сафарова оценить сроки доставки. Зная, из каких аэропортов поступают грузы, часовой пояс в каждом из них, время взлёта рейсов из портов и время полёта, определите кратчайшие сроки доставки тяжёлого самоходного геологического оборудования сотрудникам корпорации. Часовой пояс в пункте назначения *UTC +1* – один час вперёд от всемирного координированного времени.

Считается, что все самолёты с оборудованием будут взлетать хотя бы через минуту после полуночи по местному времени в аэропорту прибытия.

Формат ввода

На вход программе в первой строке подаётся натуральное число N ($N \leq 10$) – количество аэропортов. Далее в N строках для каждого аэропорта через пробел идёт несколько чисел:

целое число d_i – смещение относительно *UTC* ($-12 \leq d_i \leq 12$)

два целых числа h_i, m_i – часы и минуты взлёта по времени аэропорта, натуральное число t_i ($t_i \leq 10000$) – длительность полёта в минутах.

Формат вывода

Выведите через пробел два целых числа – через сколько часов и минут после запроса профессора Львова, приземлится последний самолёт с тяжёлым геологическим оборудованием.

Пример

Ввод	Вывод
3 -1 3 40 235 5 2 30 300 1 4 50 200	27 30

Примечание: самолёт из второго аэропорта не мог взлететь до запроса профессора.

Решение

```
n = int(input())
answer = 0

for i in range(n):
    d,h,m,t=map(int,input().split())
    uts = t+(24*60+60*h+m - 60*(d-1)) % (24*60)
    answer = max(answer,uts)

print(answer//60, answer%60)
```