

Задача 1

Предприятие непрерывного цикла «Гранулит» впервые начинает свою работу в понедельник 27 ноября 2023 года. В ходе производства некоторое количество воды объемом V литров загрязняется. Каждую среду вся загрязненная вода заливается в очистные сооружения. Цикл очистки заканчивается к утру понедельника. За это время может очиститься M литров воды. Вся очищенная вода возвращается в производство утром этого же понедельника. В очистных сооружениях может находиться не более Z литров воды. Необходимо определить на какой по счету день работы предприятия, очистные сооружения переполнятся, то есть количество воды в очистных сооружениях станет больше Z литров. 27 ноября 2023 года считать первым днем работы предприятия.

Формат ввода

На вход программе в одной строке подается три целых числа, записанные через пробел V, M, Z ($1 \leq V, M, Z \leq 10^9$).

Формат вывода

Вывести одно целое число – номер дня, в который произойдет переполнение очистных сооружений.

Если переполнение не произойдет никогда, вывести число 0.

Пример

Входные данные	Вывод	Пояснение
100 50 110	10	В среду поступит 100 литров. В понедельник уйдет 50. Останется 50 литров. В среду поступит 100 литров. Получится 150 литров, что больше 110. Произойдет переполнение. Это день номер 10 от момента начала работы предприятия.

Решение

```
V, M, Z = map(int, input().split())
k = 0
d = V - M
if V > Z:
    print(3)
elif d <= 0:
    print(0)
else:
    print (((Z - V + d) // d) * 7 + 3)
```

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 игрок пропускает ход.
6. Если игрок в своем диапазоне может найти число, сопрягаемое с обоими концами цепочки, то игра завершается его выигрышем.

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

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

Заметим, что оба числа являются трехзначными!

Например, если $N = 213$, тогда M может быть 106, 306, 506, 706 или 906. Для числа $N = 812$, число M может быть только 211, 411, 611 или 811.

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:

0, если игрок пропускает ход.

Максимальное из сопрягаемых чисел, которым игрок может завершить игру с указанием перед ним без пробела литеры “V”.

Если игрок может сделать ход, но не завершить игру, то максимальное из сопрягаемых с концами 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 **правый** конец.

Входные данные	Вывод	Примечание
812 434 400 800	V611	В диапазон игрока попало два числа, которыми можно завершить игру 411 и 611 (они сопрягаются с обоими концами цепочки). Наибольшее из них 611.
213 812 306 520	L506	На данный момент игровая цепочка выглядит: 213 ... 812 Множество сопрягаемых с концами цепочки чисел: 106, 306, 506, 706, 906, 211, 411, 611 или 811. Из них в распоряжении игрока только 3 числа: 306, 506, 411. Среди них нет числа, которое бы сопрягалось с обоими концами (игра не завершается). Наибольшее из них 506, и оно сопрягается с левым концом. После хода цепочка будет иметь вид: 506 213 ... 812
213 812 350 400	0	Множество сопрягаемых с концами цепочки чисел: 106, 306, 506, 706, 906, 211, 411, 611 или 811. К сожалению, в диапазон игрока не входит ни одно из чисел, поэтому он пропускает ход.

Решение

```
def linking(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 = [x for x in linking(left) if a <= x <= b]
mR = [x for x in linking(right) if a <= x <= b]
res = [x for x in mL if x in mR]
hod = mL + mR
print(res, hod)

if not hod:
    print(0)
elif res:
    mn = max(res)
    print("V" + str(mn))
else:
    mn = max(hod)
```

```

if mn in mR:
    print("R" + str(mn))
elif mn in mL:
    print("L" + str(mn))

```

Задача 3

Шпунтик очень любил мастерить разные вещи. Однажды он задумал соорудить машину для уборки улицы. Он трудился целую неделю и создал автомобиль "Винтик-4". Шпунтик запрограммировал автомобиль на выполнение нескольких команд:

- **Up** – вызывает *заккрытие мусоросборника*, то есть исполнитель перестает собирать мусор. Выполнение команды не изменяет заряда исполнителя;
- **Down** – вызывает *открытие мусоросборника*, то есть исполнитель продолжает собирать мусор. Выполнение команды не изменяет заряда исполнителя;
- **Left n** - вызывает изменение направления движения на 90 градусов *влево* и перемещение исполнителя на *n* единиц расстояния. Команда уменьшает заряд исполнителя на *f(n)* единиц.
- **Right n** - вызывает изменение направления движения на 90 градусов *вправо* и перемещение исполнителя на *n* единиц расстояния. Команда уменьшает заряд исполнителя на *f(n)* единиц.

Функция $f(n) = \frac{n \cdot (n+1)}{2} + 1$

Испытание своего изобретения запланировал на специальном полигоне, разлинованном на единичные клетки. В узлы всех "особых" клеток Шпунтик положил по 1 фантику. Автомобиль разместил в точке с координатами (0,0) вдоль положительного направления оси абсцисс.

Определите, сколько фантиков собрал бы исполнитель Винтик-4, если известны: начальный заряд исполнителя и список команд.

Шпунтик решил, что клетка с ненулевыми координатами (x,y) считается "особой", если $НОД(|x|,|y|) = 1$ (НОД – наибольший общий делитель чисел)

Примеры:								
(x,y)	(5,1)	(-7,3)	(-11,-1)	(4,2)	(-21,7)	(1,0)	(-5,0)	(0,0)
"особая"	ДА	ДА	ДА	НЕТ	НЕТ	НЕТ	НЕТ	НЕТ

Считается, что команда может быть выполнена, если размера заряда хватает на её полное выполнение.

В начальный момент мусоросборник закрыт.

Если заряда не хватает на выполнение команды, то исполнитель переходит к выполнению следующей команды.

Гарантируется, что размер помещения достаточен для выполнения всех команд.

Входные данные:

В первой строке указан начальный заряд исполнителя и количество команд *N*. В следующих *N* строках записаны команды для исполнителя.

Все числа неотрицательные целые (уровень заряда не более 10⁹, остальные не более 10⁶).

Выходные данные

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

Примеры:

входные данные	Выходные данные	Пояснение
1000 9 Down Left 1 Right 5 Right 3 Right 4 Right 5 Right 2 Right 4 Right 3	16	<p> Путь автомобиля представлен ломаной ABCDEFGHI, "особые" точки (точки с фантиками) отмечены оранжевыми крестами. Заряда хватит на выполнение всех команд. Автомобиль проедет через 16 различных "особых" точек (в точках (1,1), (3,1), (1,-1) он побывает дважды). Автомобиль соберёт все фантики по маршруту (мусоросборник был открыт 1-й командой и не закрывался) </p>
1000 2 Left 1 Right 10	0	На протяжении всего маршрута мусоросборник был закрыт
1000 3 Down Left 10 Right 1	1	На протяжении всего маршрута была только одна особая точка
1000 7 Down Left 3 Right 100 Left 1 Up Left 3 Left 1	1	На выполнение команды "Right 100" недостаточно энергии, и она будет пропущена. Часть фантиков на маршруте не будут собраны из-за подъёма мусоросборника

Решение

```

def fnod(a,b):

```

```

if a and b :
    a,b=abs(a),abs(b)
    while b!=0 :
        a,b=b,a%b
    return a
return 0
def stz(nn):
    return nn*(nn+1)//2+1

zz,m=map(int,input().split())
Sdvig=[(1,0),(0,1),(-1,0),(0,-1)]
tt=set()
x,y,curs=0,0,0
down=False
k=0
for _ in range (m):
    ss=input()
    if 'Up' in ss :
        down=False
        continue
    if 'Down' in ss :
        down=True
        continue
    s,n=ss.split()
    n=int(n)
    z0=stz(n)
    if zz<z0 : continue
    k+=1
    zz-=z0
    if s == 'Left' : curs=(curs+1)%4
    elif s == 'Right' : curs=(curs+3)%4
    a,b=Sdvig[curs]
    if (down==False) or (x==0 and a==0) or (y==0 and b==0) :
        x+=a*n
        y+=b*n
        continue
    for i in range(n):
        x+=a
        y+=b
        d=fnod(x,y)
        if d==1 :
            tt.add((x,y))
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. Процедура повторяется, пока в стопке есть карточки. Какое наибольшее четырёхзначное число можно собрать таким образом, если известна последовательность карточек?

Формат ввода

На вход программе в первой строке подается натуральное число N ($4 \leq N \leq 10000$) – 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 7 2 3 6 9	9763

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». Миша решил передавать роботу инструкции в виде цифр числа: робот получает число, переводит его в двенадцатеричную систему и выполняет соответствующие цифрам команды. Коды команд, отвечающих за перемещение, кратны трём.

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 33 130 144 12	4

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

Группа российских прикладных востоковедов испытывает в верховьях Евфрата новое научное оборудование, разработанное в МГТУ. Для успешной работы им требуется радиолокационное покрытие. Доцент Фабов обозначил ряд позиций, на которые могут выехать мобильные локаторы. Для эффективной работы расстояние между локаторами должно быть не меньше определённого значения. Позиции располагаются на луче, выходящем из лагеря востоковедов. Располагая списком расстояний от позиций до лагеря, отсортированным по возрастанию, определите, какое максимальное количество локаторов можно разместить на этих позициях.

Формат ввода

На вход программе в первой строке подаётся натуральное число N ($N \leq 1000$) – количество позиций. Во второй строке подаётся натуральное число L ($L \leq 100$) – минимально допустимое расстояние между локаторами. Далее в N строках идёт по одному натуральному числу d_i – расстояние позиции номер i от лагеря востоковедов ($1 \leq d_i \leq 1000$). Все расстояния отсортированы по возрастанию.

Формат вывода

Выведите одно целое число – какое максимальное количество локаторов можно разместить на этих позициях,

Пример

Ввод	Вывод
5 20 1 21 32 43 45	3

Решение

```
def work(dat,l):  
    k = 1  
    temp = dat[0]  
    for i in range(1,len(dat)):  
        if dat[i]>=temp+l:  
            k+=1  
            temp = dat[i]  
    return k
```

```
n = int(input())  
l = int(input())
```

```
answer = 0
data = n*[0]
for i in range(n):
    data[i] = int(input())

answer = 0
for i in range(len(data)):
    answer = max(answer,work(data[i:],1))

print(answer)
```