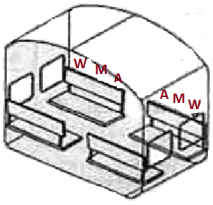
Кузнецов Максим Алексеевич

Балл: 105

Задача №1

Условие

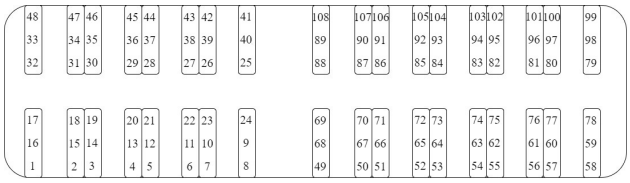
Миша и **Маша** очень любят путешествовать. В основном они путешествуют по железной дороге. Однажды они ехали в вагоне, где были только сидячие места, и заинтересовались расположением сидений в своем купе. Купе выглядело примерно так



Тип места обозначается следующим образом:

- Сиденье у окна: **W**
- Среднее место: **M**
- Место у прохода: **A**

Нумерация мест в вагоне приведена на следующем рисунке



Ребятам стало интересно узнать номер и тип сиденья, расположенного рядом с ними. В случае, когда сиденье расположено в середине, необходимо указать оба соседних номера.

Напишите программу, которая поможет ребятам решить эту задачу.

Входные данные

На вход подается одно целое число **N (**1 <= **N** <= 108**),** обозначающее номер места.

Выходные данные

Целое число, и большая латинская буква **W, M** или **A**, разделенные одним пробелом, которые будут соответствовать номеру и типу соседнего сиденья. Если таких сидений два, то данные по второму сидению выводится в той же строке через пробел (номера сидений в этом случае выводятся в порядке возрастания).

Примеры

Входные данные	Выходные данные
19	14 M
96	81 A 101 W

Исходный код

```
arr = [[1,16,17],[2,15,18],[3,14,19],[4,13,20],[5,12,21],[6,11,22],[7,10,23],[8,9,24],[49,68,69],[50,67,70],[51,66,71],[52,65,72],[53,64,73],[54,63,74],[55,62,75],[56,61,76],[57,60,77

sit = int(input())

for count1 in range(len(arr)):
    for count2 in range(3):
        if arr[count1][count2] == sit:
            if (count2 == 0):
                print(arr[count1][1], 'M')
            if count2 == 1:
                print(arr[count1][2], 'A', arr[count1][0],'W')
            if count2 == 2:
                print(arr[count1][1], 'M')
```

Задача №2

Условие

Петя наблюдает, как строители копают фундамент для нового дома. Землю они вывозят одинаковыми самосвалами. По каким-то своим причинам всю выбранную землю вывозят со стройки в тот же день, в который ее выбрали, то есть даже за одним кубометром все равно вечером приедет самосвал.

Пете известно, сколько времени копали фундамент, какой объем земли вывозили каждый день и сколько кубометров помещается в самосвал. Он задался вопросом: насколько меньше рейсов понадобилось бы совершить, если бы можно было вывозить землю не в тот же самый день, в который ее извлекли?

Напишите программу, которая решит эту задачу за Петю.

В первой строке на вход программы подаются два натуральных числа: количество дней t, в течение которых копали фундамент, $1 < t \leq 10^3$ и объем земли v, который вмещается в один самосвал, $1 < v \leq 10^2$. На следующей строке на вход программы поступает n натуральных чисел: $a_1, a_2, \ldots, a_n$. a_i – сколько земли выбрали в i-й день, $1 < a_i \leq 10^5$.

Выведите одно целое число – разницу между количеством проделанных рейсов и минимальным количеством рейсов, за которые можно вывезти этот объем земли.

Пример

Входные данные	Выходные данные
2 4 6 5	1

Комментарий к примеру. Строители работали два дня, в первый день они извлекли 6 кубометров грунта, во второй – 5. Самосвал принимает 4 кубометра грунта, следовательно, понадобится 4 рейса. Можно было обойтись за три рейса, следовательно, разница – 1.

Исходный код

```
arr = input()
arr = list(map(int,arr.split())) #Вводим первую строку

time = arr[0]
v = arr[1]

arr = input()
arr = list(map(int,arr.split())) #Вводим вторую строку

quantity_Iterations = 0

for counter in arr: #Перебираем все дни
    quantity_Iterations += counter // v #Считаем сколько раз он съездил
    if (counter % v) > 0: #Если еще остался грунт
        quantity_Iterations += 1 #Добавляем еще один раз

sum_V = 0

for counter in arr: #Перебираем все дни
    sum_V += counter #Считаем общий объем

razn = quantity_Iterations - (sum_V // v)

dop = 0

if (sum_V % v) > 0:
    dop = 1

print(razn - dop)
```

Задача №3

Условие

Исполнитель получает на вход натуральное число X (не превышающее 10^6). По этому числу, точнее по его представлению в шестеричной системе счисления, строится новое число Y по следующим правилам.

В шестеричном представлении числа X предпоследняя цифра увеличивается на 1 (гарантируется, что в шестеричном представлении X числа больше 2-х цифр). Например, $749_{10} = 32\mathbf{45}_6 \to 32\mathbf{55}_6 = 755_{10}$.

Если предпоследняя цифра 5, тогда предпоследняя цифра становится 0, а последняя изменяется по следующему принципу: четная увеличивается н 1, а нечетная уменьшается на 1. Например, последняя цифра нечетная $751_{10} = 32\mathbf{51}_6 \to 32\mathbf{00}_6 = 720_{10}$, последняя цифра четная $752_{10} = 32\mathbf{52}_6 \to 32\mathbf{03}_6 = 723_{10}$.

Введем понятие расстояния

Oh = Исходное_число – Полученное_число

Напишите программу, которая будет считать для скольких чисел из заданного интервала [A,B] расстояние Oh будет отрицательным.

На вход программы подаётся два целых числа A и B ($10 \le A \le B \le 1\,000\,000$), записанных через пробел.

Программа должна вывести одно целое число - вычисленное значение.

Входные данные	Вывод	Примечание
748 752	2	748 – 754 < 0 ($32\mathbf{44}_6 – 32\mathbf{54}_6$) 749 – 755 < 0 ($32\mathbf{45}_6 – 32\mathbf{55}_6$) 750 – 721 > 0 ($32\mathbf{50}_6 – 32\mathbf{01}_6$) 751 – 720 > 0 ($32\mathbf{51}_6 – 32\mathbf{00}_6$) 752 – 723 > 0 ($32\mathbf{52}_6 – 32\mathbf{03}_6$)

Исходный код

```
def to_Six(num):
    arr_Num = []
    while num > 1:
        arr_Num.insert(0,num % 6)
        num = num // 6

    return arr_Num


def to_Ten(arr_Num):
    num = 0

    for counter in range(len(arr_Num)):
        num += arr_Num[len(arr_Num) - counter - 1] * (6 ** counter)

    return num


def new_Calc(num):

    arr_Num = to_Six(num)

    if arr_Num[len(arr_Num) - 2] != 5:
        arr_Num[len(arr_Num) - 2] += 1
    else:
        arr_Num[len(arr_Num) - 2] = 0
        if (arr_Num[len(arr_Num) - 1] % 2) == 0:
            arr_Num[len(arr_Num) - 1] += 1
        else:
            arr_Num[len(arr_Num) - 1] -= 1
    return to_Ten(arr_Num)

arr = input()
arr = list(map(int,arr.split()))

count = 0
```

```
for counter in range(arr[0],arr[1]):
    if counter - new_Calc(counter) < 0:
        count += 1
print(count)
```

Задача №4

Условие

Зонд передает данные с орбиты Юпитера во время сильной магнитной бури. Информация передается по каналу связи в виде пакетов. Каждый пакет представляет собой целое положительное число в двоичной системе счисления. Для обнаружения помех последний разряд в пакете подбирают таким образом, чтобы количество единиц в разрядах пакета было четным. В каждом пакете на практике никогда не искажается больше одного разряда.

Напишите программу, которая по распечатке пакетов, записанных в десятичной системе счисления, найдет самое большое значение, прошедшее без искажений. Известно, что как минимум один пакет прошел без искажений.

Формат ввода

В строке вводится сначала целое число ***n*** – количество пакетов (***n ≤ 1000***), затем ***n*** натуральных чисел, все числа отделены друг от друга одним или несколькими пробелами.

Формат вывода

Вывести одно целое число – самое большое значение, прошедшее без искажений.

Пример

Входные данные	Выходные данные
4 1025 496 882 1056	1056

Исходный код

```
def to_Two(num):
    arr_Num = []
    while num > 1:
        arr_Num.insert(0,num % 2)
        num = num // 2
    arr_Num[0] = 1
    return arr_Num


arr = input()
arr = list(map(int,arr.split()))

max_Num = 0

for counter in arr:
    arr_Num = to_Two(counter)
    if ((arr_Num.count(1) % 2) == 0) and (counter > max_Num):
        max_Num = counter

print(max_Num)
```

Задача №5

Условие

Вася придумывает пароль для каждой новой учетной записи, которую он заводит на каком-то из своих устройств. Время от времени он изменяет пароль, дописывая к нему новые символы. Предыдущий пароль никогда не будет началом пароля для новой учетной записи Васи. После того, как Вася заводит новую учетную запись, он перестает менять пароль на старой.

Зная все Васины пароли в хронологическом порядке, напишите программу, которая найдет, какое наибольшее количество раз Вася менял пароль для одной учетной записи.

Формат ввода

В первой строке вводится сначала целое число ***n*** – количество слов (***n ≤ 1000***), затем в ***n*** следующих строках записано по слову. Слова состоят только из строчных латинских букв.

Формат вывода

Вывести одно целое число – какое наибольшее количество раз Вася менял пароль для одной учетной записи.

Примеры

Входные данные	Выходные данные
3 abd abdc bvd	1
4 bcd bcdf bcdfe abc	2

Исходный код

```
quantity_Iterations  = int(input())

password = input()
pre_Pass = password

counter = 0
max_Counter = 0

for i in range(quantity_Iterations - 1):
    password = input()
    if pre_Pass == password[:len(pre_Pass)]:
        counter +=1
    else:
        if counter > max_Counter:
```

```
            max_Counter = counter

        counter = 0

print(max_Counter)
```

Задача №6

Условие

Миша ведет дневник, отслеживая, сколько шагов в день он проходит пешком. Каждый день он записывает количество шагов. Требуется определить, в какой день во время периода наблюдений он прошел наибольшую дистанцию и прошел ли он столько, сколько планировал, за все время.

На вход программе в первой строке подаются натуральное число N – количество дней, в течение которых велось наблюдение ($N \leq 20$) и натуральное число X – запланированное количество шагов.

Далее в N строках подается на вход по целому положительному числу – x_i: количество шагов, пройденных в i-й день.

Вывести два целых числа – в первой строке вывести номер дня, в который Миша прошел больше всего шагов, во второй строке вывести 1, если он прошел не менее запланированного числа шагов и 0 в обратном случае. Если Вася прошел одинаковое количество шагов в несколько разных дней, вывести наименьший номер дня.

Пример

Ввод	Вывод
3 10000 5100 6500 4200	2 1

Исходный код

```
arr = input()
arr = list(map(int,arr.split()))

days = arr[0]
plan = arr[1]

max_Steps = 0

summ_Steps = 0

max_Day = 0

for i in range(days):
    steps = int(input())
    if steps > max_Steps:
        max_Steps = steps
        max_Day = i + 1

    summ_Steps += steps

print(max_Day)

if summ_Steps >= plan:
    print(1)
else:
    print(0)
```

Задача №7

Условие

Исследовательский аппарат на поверхности Марса может выполнять команды «Фотографирование», «Пробное бурение», «Взятие образцов грунта», «Анализ атмосферы». Из-за конструктивных особенностей на аппарат наложен ряд ограничений. Нельзя выполнять команду «Анализ атмосферы» после команды «Пробное бурение». Нельзя выполнить команду «Фотографирование» после команды «Взятие образцов грунта». Команду «Взятие образцов грунта» можно выполнять только следующей после команды «Пробное бурение». Никакую команду, кроме команды «Взятие образцов грунта», нельзя выполнить подряд дважды.

Напишите программу, которая определит, сколько существует выполнимых последовательностей команд длиной n, если до начала выполнения программы аппарат выполнил команду «Пробное бурение».

На вход программе подается натуральное число n ($n \leq 15$) – количество команд.

Вывести целое число – количество выполнимых последовательностей команд длиной n.

Пример

Ввод	Вывод
2	5

Исходный код

```
def calc(num):
    if num == 0:
        return [1,3]
    if num == 1:
        return [0,2]
    if num == 2:
        return [1,2,3]
    if num == 3:
        return [0,1]


def var_Calc(arr):
    new_Arr = []
    for i in arr:
        new_Arr += calc(i)
    return new_Arr

quant_Comand = int(input())

arr = [1]

for i in range(quant_Comand):
    arr = var_Calc(arr)

print(len(arr))
```