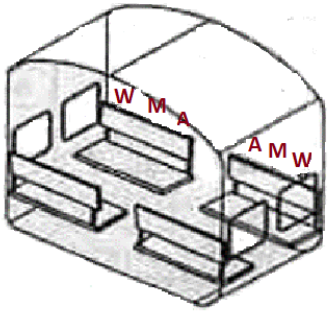
Перов Платон Викторович

Балл: 120

Задача №1

Условие

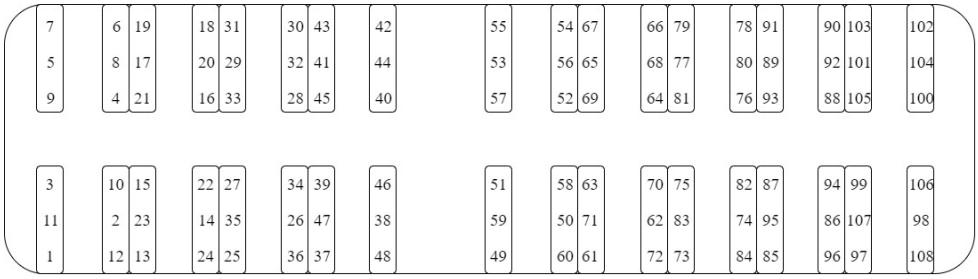
Витя и **Оля** очень любят путешествовать. В основном они путешествуют по железной дороге. Однажды они ехали в вагоне, где были только сидячие места, и заинтересовались расположением сидений в своем купе. Купе выглядело примерно так



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** и л и **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.

Примеры

Входные данные	Выходные данные
19	17 M
29	31 W 33 A

Исходный код

```
train = [[1, 11, 3, 9, 5, 7],
[12, 2, 10, 4, 8, 6],
[13, 23, 15, 21, 17, 19],
[24, 14, 22, 16, 20, 18],
[25, 35, 27, 33, 29, 31],
[36, 26, 34, 28, 32, 30],
[37, 47, 39, 45, 41, 43],
[48, 38, 46, 40, 44, 42],
[49, 59, 51, 57, 53, 55],
[60, 50, 58, 52, 56, 54],
[61, 71, 63, 69, 65, 67],
[72, 62, 70, 64, 68, 66],
[73, 83, 75, 81, 77, 79],
[84, 74, 82, 76, 80, 78],
[85, 95, 87, 93, 89, 91],
[96, 86, 94, 88, 92, 90],
[97, 107, 99, 105, 101, 103],
[108, 98, 106, 100, 104, 102]]
n = int(input())
cur = []
for i in train:
    if n in i:
        cur = i
        break
if cur.index(n) == 0:
    print(cur[1], 'M')
elif cur.index(n) == 1:
    print(cur[0], 'W', end=' ')
    print(cur[2], 'A')
elif cur.index(n) == 2:
    print(cur[1], 'M')
elif cur.index(n) == 3:
    print(cur[4], 'M')
elif cur.index(n) == 4:
    print(cur[5], 'W', end=' ')
    print(cur[3], 'A')
else:
    print(cur[4], 'M')
```

Задача №2

Условие

В кассе, продающей билеты по цене 500 р., нет возможности расплатиться картой, а в начале работы нет даже денег для сдачи. Перед кассой стоит очередь, у каждого человека в очереди есть только одна купюра. У части – 500 р., у части – 1000 р. Сколькими способами можно выстроить очередь так, чтобы у кассира всегда была сдача с тысячерублевой купюры?

Напишите программу, которая решит эту задачу.

Сначала на вход программы подаются два натуральных числа: количество человек $\boldsymbol{n}$ с купюрой 500 р, $1 < \boldsymbol{n} \leq 18$ и количество человек с купюрой 1000 р. $\boldsymbol{k}$, $1 < \boldsymbol{k} \leq 18$.

Выведите одно целое число – сколькими способами можно выстроить очередь.

Пример

Входные данные	Выходные данные
2 2	2

Исходный код

```
from math import factorial
n, k = map(int, input().split())
if k > n:
    print(0)
else:
    res = factorial(n + k) // factorial(n) // factorial(k) - factorial(n + k - 1) // factorial(n) // factorial(k - 1)
    for i in range(2, k + 1):
```

```
        res -= (i - 1) * factorial(n - i + 1 + k - i) // factorial(n - i + 1) // factorial(k - i)
    print(res)
```

Задача №3

Условие

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

В девятеричном представлении числа X зачеркиваются последние две цифры (гарантируется, что в девятеричном представлении числа X больше 2-х цифр).

1. Если получившееся число четное, то в конец полученной цепочки цифр дописывается две цифры – сумма оставшихся цифр (тоже в девятеричном представлении, причем если сумма состоит из одной цифры, то перед ней надо вписать 0).

   Например, $4862_{10} = 6602_7$. После зачеркивания остается 66_9 – четное число. Сумма оставшихся цифр $12_{10}=13_9$. Новое число Y = $6613_9 = 4872_{10}$.

   Другой пример, $3140_{10} = 4268_9$, после зачеркивания остается четное число 42_9. Сумма оставшихся цифр $6_{10} = 6_9$. Новое число Y= $42\mathbf{0}6_9 = 3084_{10}$.

2. Если получившееся после зачеркивания последних двух цифр число нечетное, то в конец полученной цепочки цифр дописывается одна цифра – количество оставшихся цифр. Например, $4859_{10} = 6588_9$. После зачеркивания остается 65_9 – нечетное число. Количество оставшихся цифр $2_{10}=2_9$. Новое число Y = $652_7 = 533_{10}$.

Напишите программу, которая будет искать наибольшее число, которое получается на заданном интервале [A,B], и наибольшее число, из которого оно было получено.

На вход программы подаётся два целых числа A и B ($90 \le A \le B \le 1\,000\,000$), записанных через пробел.

Программа должна вывести два числа наибольшее полученное число и через пробел число, из которого оно было получено исполнителем.

Входные данные	Вывод	Примечание
4859 4862	4872 4862	4859 -> 533 4860 -> 4872 4861 -> 4872 **4862 -> 4872** Наибольшее значение, получаемое на заданном интервале 4872. Оно получается из трех чисел 4860, 4861 и 4862. Наибольшее из них 4862.

Исходный код

```
def func(x):
    s = ''
    cnt = 0
    while x:
        cnt += x % 9
        s += str(x % 9)
        x //= 9
    s = s[::-1]
    cnt -= int(s[-1]) + int(s[-2])
    s = s[:-2:]
    if int(s[-1]) % 2 == 0:
        if cnt < 9:
            s += '0' + str(cnt)
        else:
            s += str(cnt  // 9)  + str(cnt % 9)
    else:
        s += str(len(s))
```

```
    s = int(s, 9)
    return s
a, b = map(int, input().split())
res1 = -1
res2 = -1
cnt = 0
for i in range(max(a, b - 100000), b + 1):
    res = func(i)
    if res >= res1:
        res1 = res
        res2 = i


print(res1, res2)
```

Задача №4

Условие

Передатчик отправляет по одному слову в секунду. Слово состоит из строчных латинских букв. Сообщение состоит из нескольких слов. Слова в одном сообщении заканчиваются на одну букву. Напишите программу, которая найдет сообщение, содержащее больше всего слов.

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 – сколько слов было в искомом сообщении.

Примеры

Входные данные	Выходные данные
3 abc bac sdfe	2
4 me pudge purge tou	3

Исходный код

```
n = int(input())
alph = [0] * 26
for i in range(n):
    s = input()
    alph[ord(s[-1]) - 97] += 1
print(max(alph))
```

Задача №5

Условие

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

```
a = list(map(int, input().split()))
n = a[0]
res = -1
for i in range(1, n + 1):
    s = bin(a[i])[2:]
    if s.count('1') % 2 == 0:
        res = max(res, a[i])
print(res)
```

Задача №6

Условие

Для обогрева пассажирского отсека вездехода «Химера» необходимо подключить нагревательный элемент. Элемент питается от сети постоянного тока напряжением U и состоит из двух последовательно соединенных резисторов, сопротивление каждого из которых не должно быть меньше порогового значения R. Напишите программу, которая найдет в списке резисторов пару, которая обеспечит наибольшую мощность элемента. Гарантируется, что хотя бы два подходящий нагревателя в списке есть.

На вход программе в первой строке подаются натуральное число N – количество резисторов (N≤20) и вещественное положительное число R – ограничение по мощности.

Далее в N строках подается на вход по целому положительному числу – r_i: сопротивление i-го резистора.

Вывести одно целое число – сопротивление нагревательного элемента.

Мощность рассчитывается по формуле

$$P = \frac{U^2}{R}$$

где U – напряжение, R – сопротивление. При последовательном соединении сопротивления складываются.

Пример

Ввод	Вывод
3 5 5 6 7	11

Исходный код

```
n, r = map(float, input().split())
n = int(n)
a = [int(input()) for i in range(n)]
res = max(a) * 4
for i in range(n):
    for j in range(n):
        if i != j and a[i] >= r and a[j] >= r and a[i] + a[j] < res:
            res = a[i] + a[j]
print(res)
```

Задача №7

Условие

В пошаговой браузерной игре игрок командует космическим фрегатом. Во время боя у него есть выбор из нескольких команд: «Поднять щиты», «Ракетный залп», «Орудийный залп» и «Маневр уклонения». Каждый ход можно выполнить только одну команду. Из-за игровой механики нельзя выполнить команду «Орудийный залп» в следующий ход после «Маневра уклонения». Нельзя выполнить команду «Ракетный залп» в следующий ход после подъема щитов. Нельзя два хода подряд поднимать щиты.

Напишите программу, которая определит, сколько существует выполнимых последовательностей команд длиной n.

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

Вывести целое число – количество выполнимых последовательностей команд длиной n.

Пример

Ввод	Вывод
2	13

Исходный код

```
answers = [4, 13, 42, 135, 434, 1395, 4484, 14413, 46328, 148913, 478654, 1538547, 4945382, 15896039, 51094952]
n = int(input())
print(answers[n - 1])
```