Основной тур олимпиады для педагогов "Код Архимеда"

## Задача A. Рулетка, но не в Монте-Карло

Линейный поиск. Сортировка.

В задаче следует найти наибольшую совпадающую пару значений и наименьшую несовпадающую.

n = int(input())

L = list(sorted(map(int, input().split())))

M = list(sorted(map(int, input().split())))

i = 0

while L[i] == M[i]:

x = L[i]

i += 1

y = max(L[i], M[i])

if L[i] > M[i]:

ans = 1

else:

ans = 2

print(ans, x, y)

## Задача B. Пирамиды, но не в Египте

Циклы, проверка условий.

На первый взгляд количество невырожденных пирамид ограничивается только неравенством треугольника: нам подойдут все значения ***x*** и ***y*** от 1 до 2 × ***z*** - 1. Но есть еще случай, когда пирамида вырождается в плоскость, например, при ***x*** = 6, ***y*** = 8 и ***z*** = 5 она превращается в четыре прямоугольных пифагоровых треугольников. Эти случаи нужно проверять и не учитывать.

z = int(input())

ans = 0

for i in range(1, 2 \* z):

for j in range(i, 2 \* z):

ans += 1

if i % 2 == 0 and j % 2 == 0 and (i // 2) \*\* 2 + (j // 2) \*\* 2 == z \*\* 2:

ans -= 1

print(ans)

## Задача C. Граф, но не Монте-Кристо

Динамическое программирование.

Простая задача на реализацию одномерного динамического программирования. База динамики для города A равна 1. Далее последовательно определяем возможные способы попасть в город B, C и так далее как сумму всех вычисленных ранее значений для городов, из которых есть путь в текущий город.

n, m = map(int, input().split())

L = [[i] for i in range(n)]

for i in range(m):

a, b = input().split()

a = ord(a) - 65

b = ord(b) - 65

L[a].append(b)

dp = [0 for i in range(n)]

dp[0] = 1

for i in range(n):

for j in range(1, len(l[i])):

dp[l[i][j]] += dp[i]

print(dp[-1])