**Задача 1. (15 балів) NEIGHBORS** (*Сусіди*)

Задана матриця Neighbors, розмірності **n,** елементи якої Neighbors (i,j) рівні **1**, якщо країни **i** і **j** мають спільний кордон, і **0** - в протилежному випадку. Скласти процедуру пошуку країни, що має найбільшу кількість сусідів серед перелічених країн.

**Вхідні дані**

Перший рядок ціле число **n.**

Кожний з наступних **n** рядків містить **n** чисел.

***Вихідні дані***

*Номер шуканої країни.*

**Приклад вхідних даних:**

4

0 1 1 0

1 0 1 0

1 1 0 1

0 0 1 0

**Вихідні дані:**

3

**Код:**

n = int(input())

neighbors = []

for i in range(n):

 neighbors.append(list(map(int, input().split())))

m = [0] \* n

for i in range(n):

 m[i]+=neighbors[i].count(1)

print(m.index(max(m))+1)

**Задача 2. На уроці (25 балів)**

На уроці іноземної мови вивчають англійську абетку. Учителька задає текст **S** , це сукупність слів *розділених не менше, як одним пропуском*, а учні повинні переставити літери кожного слова з цього тексту за абеткою (від «А» до «Z»). Щоб перевіряти себе, Андрійко хоче мати програму, яка це робить, складіть дня нього таку програму.

**Вхідні дані:**

Рядок, у якому міститься текст **S**.

**Вихідні дані:**

Рядок, у якому міститься текст, що утворився після перестановки літер.

**Зауваження:** *слова в стрічці S розділені не менше, як одним пропуском, тільки великі латинські літери входять в слова.*

*Приклад вводу*:

he is a mathematician

HE IS A MATHEMATICIAN

*Приклад виводу*:

EH IS A AAACEHIIMMNTT

Код:

st = input().split()

for i in range(len(st)):

 st[i] = list(st[i])

 st[i].sort()

 st[i] = ''.join(st[i])

print(' '.join(st))

**Задача 3. " Призери олімпіади" (60 балів).**

Організатори олімпіади з програмування, вирішили нагородити її переможців, але в цікавий спосіб. Призерам вручили по N талонів , на кожному із яких вказана сума sk ( в гривнях), на яку він може обміняти даний талон, здавши його в оргкомітет на першій хвилині і вказано також час tk (tk ≤ 30) хвилин (час протягом якого талон дійсний), при цьому оголошено, що вартість кожного із талонів зменшується з кожною наступною хвилиною від початку старту на суму, пропорційну його вартості і часу (Sk /tk) і протягом хвилини можна обміняти тільки один талон. Наприклад, якщо талон вартує 200 гр., а час протягом якого талон дійсний 5 хв., то обмінявши його на першій хвилині після старту, учень отримає 200 гр., а , якщо він його обміняє на другій хвилині, то вже отримає тільки 160 гр. (200 - 200/5), на третій –120 грн. Чи може учень придбати за отримані кошти комп’ютер вартість якого С. Старт для обміну талонів дається через певний час після оголошення інформації про умови отримання винагороди. За даний час учень має можливість за комп’ютером скласти програму, яка підкаже йому у якому порядку обмінювати талони, щоб отримати максимальну суму коштів, і дає відповідь, чи можна придбати за ці кошти комп’ютер.

**Вхідні дані**

Перший рядок містить кількість талонів N і ціну комп'ютера  С.

Кожний з наступних N рядків містить два числа – кількість хвилин tk і вартість  Sk.

**Вихідні дані**

Файл містить інформацію «Так» або «Ні».

**Приклад**

**Вхідні дані Вихідні дані**

4 230 Так

2 148

3 135

1 46

4 120

4 2300 Так

2 1480

3 1350

1 460

4 1200

**Код:**

def NextSet(A, n):

 j = n - 2

 while j != -1 and A[j] >= A[j + 1]:

 j -= 1

 if j == -1:

 return (False, A)

 k = n - 1

 while A[j] >= A[k]:

 k -= 1

 A[j], A[k] = A[k], A[j]

 s = A[j+1:]

 s.reverse()

 A = A[:j+1] + s

 return (True, A)

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

G = []

A = []

m = 0

for i in range(n):

 A.append(i+1)

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

 j = 1

 if k[0] > m:

 m = k[0]

 mas = []

 rizn = k[1]/k[0]

 while k[1] != 0:

 mas.append(k[1])

 k[1] = k[1] - rizn

 G.append(mas)

mxm = 0

Perm = []

Perm.append(A[:])

flag = True

(flag, A) = NextSet(A, len(A))

while flag == True:

 Perm.append(A[:])

 (flag, A) = NextSet(A, len(A))

mxm\_arr = []

for i in range(len(Perm)):

 sum = 0

 for j in range(m):

 if j < len(Perm[i]) and j < len(G[Perm[i][j] -1]):

 sum += G[Perm[i][j] -1][j]

 if sum > mxm:

 mxm = sum

 mxm\_arr = Perm[i]

if mxm >= C:

 print('Так')

 print(mxm\_arr)

 print(mxm)

else:

 print('Ні')