**Question no:1 Grid Challenge**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Problem

Given a square grid of characters in the range ascii[a-z], rearrange elements of each row alphabetically, ascending. Determine if the columns are also in ascending alphabetical order, top to bottom. Return YES if they are or NO if they are not.

For example, given:

a b c

a d e

e f g

The rows are already in alphabetical order. The columns a a e, b d f and c e g are also in alphabetical order, so the answer would be YES. Only elements within the same row can be rearranged. They cannot be moved to a different row.

Function Description

Complete the gridChallenge function in the editor below. It should return a string, either YES or NO.

gridChallenge has the following parameter(s):

grid: an array of strings

Input Format

The first line contains , the number of testcases.

Each of the next sets of lines are described as follows:

- The first line contains , the number of rows and columns in the grid.

- The next lines contains a string of length

Constraints

Each string consists of lowercase letters in the range ascii[a-z]

Output Format

For each test case, on a separate line print YES if it is possible to rearrange the grid alphabetically ascending in both its rows and columns, or NO otherwise.

Sample Input

------------

1

5

ebacd

fghij

olmkn

trpqs

xywuv

Sample Output

-------------

YES

Explanation

The x grid in the test case can be reordered to

abcde

fghij

klmno

pqrst

uvwxy

This fulfills the condition since the rows 1, 2, ..., 5 and the columns 1, 2, ..., 5 are all lexicographically sorted.

Sample Input-1

--------------

2

2

kc

iu

3

uxf

vof

hmp

Sample Output-1

---------------

YES

NO

Sample Input-2

--------------

3

3

abc

lmp

qrt

3

mpxz

abcd

wlmf

4

abc

hjk

mpq

rtv

Sample Output-2

---------------

YES

NO

YES

**Answer:**

for \_ in range(int(input())):

n = int(input())

matrix = []

flag = 0

for i in range(n):

matrix.append(sorted(list(input())))

for i in zip(\*matrix):

if(list(i) != sorted(list(i))):

flag = 1

break

print("YES" if flag==0 else "NO")

**Question no:2 Bank Compare**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

There are two banks – Bank A and Bank B. Their interest rates vary. You have received offers from both banks in terms of the annual rate of interest, tenure, and variations of the rate of interest over the entire tenure.You have to choose the offer which costs you least interest and reject the other. Do the computation and make a wise choice.

The loan repayment happens at a monthly frequency and Equated Monthly Installment (EMI) is calculated using the formula given below :

EMI = loanAmount \* monthlyInterestRate / ( 1 – 1 / (1 + monthlyInterestRate)^(numberOfYears \* 12))

Constraints:

1 <= P <= 1000000

1 <=T <= 50

1<= N1 <= 30

1<= N2 <= 30

Input Format:

First line: P principal (Loan Amount)

Second line: T Total Tenure (in years).

Third Line: N1 is the number of slabs of interest rates for a given period by Bank A. First slab starts from the first year and the second slab starts from the end of the first slab and so on.

Next N1 line will contain the interest rate and their period.

After N1 lines we will receive N2 viz. the number of slabs offered by the second bank.

Next N2 lines are the number of slabs of interest rates for a given period by Bank B. The first slab starts from the first year and the second slab starts from the end of the first slab and so on.

The period and rate will be delimited by single white space.

Output Format: Your decision either Bank A or Bank B.

Explanation:

Example 1

Input

10000

20

3

5 9.5

10 9.6

5 8.5

3

10 6.9

5 8.5

5 7.9

Output: Bank B

Example 2

Input

500000

26

3

13 9.5

3 6.9

10 5.6

3

14 8.5

6 7.4

6 9.6

Output: Bank A

**Answer:**

p = int(input())

t = int(input())

ans = []

for \_ in range(2):

cost = 0

for i in range(int(input())):

y, r = map(float, input().split())

cost += p\*r/(1-1/((1+r)\*\*(y\*12)))

ans.append(cost)

if(ans[0]<ans[1]):

print("Bank A")

else:

print("Bank B")

**Question no:3 Counting Rock Samples**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Juan Marquinho is a geologist and he needs to count rock samples in order to send it to a chemical laboratory. He has a problem: The laboratory only accepts rock samples by a range of its size in ppm (parts per million).

Juan Marquinho receives the rock samples one by one and he classifies the rock samples according to the range of the laboratory. This process is very hard because the number of rock samples may be in millions.

Juan Marquinho needs your help, your task is to develop a program to get the number of rocks in each of the ranges accepted by the laboratory.

Input Format:

An positive integer S (the number of rock samples) separated by a blank space, and a positive integer R (the number of ranges of the laboratory); A list of the sizes of S samples (in ppm), as positive integers separated by space R lines where the ith line containing two positive integers, space separated, indicating the minimum size and maximum size respectively of the ith range.

Output Format:

R lines where the ith line containing a single non-negative integer indicating the number of the samples which lie in the ith range.

Constraints: 10 ? S ? 10000 1 ? R ? 1000000 1?size of each sample (in ppm) ? 1000

Example 1

Input: 10 2

345 604 321 433 704 470 808 718 517 811

300 350

400 700

Output: 2 4

Explanation:

There are 10 samples (S) and 2 ranges ( R ). The samples are 345, 604,811. The ranges are 300-350 and 400-700. There are 2 samples in the first range (345 and 321) and 4 samples in the second range (604, 433, 470, 517). Hence the two lines of the output are 2 and 4

Example 2

Input: 20 3

921 107 270 631 926 543 589 520 595 93 873 424 759 537 458 614 725 842 575 195

1 100

50 600

1 1000

Output: 1 12 20

Explanation:

There are 20 samples and 3 ranges. The samples are 921, 107 195. The ranges are 1-100, 50-600 and 1-1000. Note that the ranges are overlapping. The number of samples in each of the three ranges are 1, 12 and 20 respectively. Hence the three lines of the output are 1, 12 and 20

**Answer:**

s, r = map(int, input().split())

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

for \_ in range(r):

x, y = map(int, input().split())

count = sum([1 for i in arr if i>=x and i<=y])

print(count, end=" ")