Welcome to Codefest 2.0 Fall 2022

"First, solve the problem. Then, write the code."
– John Johnson

Junior Programming Competition

Organized by GDSC
FCIT
University of the Punjab

Problem A

Record Sorting!

FCIT administration is launching a record sorting system, but they are stuck on a specific problem that involves strings to be reformatted for better record storage. Think of them as secret codes that you need to crack. You have been given a special tool, that can perform two types of operations on the first string:

1. Append a letter to the end of the string.
2. Delete the last character of the string. This operation is legal on empty strings (results in an empty string)

The strings will be lowercase. Your task is to determine whether you can convert the first string to the second string by performing exactly 'x' number of the above operations on the first string. If it is possible, return 'Yes'. Otherwise, return `'No'`.

For example, if the first string is `'abc'` and the second string is `'def'` with 6 moves available... You can delete the letters `'a'`, `'b'` and `'c'` and add the letters using 3 moves, and then add `'d'`, `'e'`, and `'f'` in that order using 3 moves. With each operation, you will get closer and closer to the second string. In the end, you'll have the matching string, and you'll know you've succeeded when you see `'Yes'` on the screen!"

Input format

```
The first line contains a string a, the initial string.
The second line contains a string b, the desired final string.
The third line contains an integer x, the number of operations.
```

Output format:

```
Yes/No
```

Sample Test Case 1

```
aba
aba
7
```

Sample Output 1

```
Yes
```

Sample Test Case 2

```
ashley
ash
2
```

Sample Output 2

```
No
```

Problem B

Cut the Pizza!

FCIT finally decides to give a farewell party to F17. This has been long due to COVID 19 restrictions. The students of F17 have been eager for this party for over a year. For this purpose, FCIT hired a special pizza company that made one large pizza for everyone. Now the problem is distributing this pizza equally to everyone.

The pizza is circular, with a bunch of pepperonis around the outside. However, while the cheese is uniform, the toppings are not: there are various toppings, and each student must receive them in the same order. Because the EMS society despises waste and will not allow leftovers, make sure you can serve the entire pizza.

To make it easier for you to cut the pizza, you have made a string out of the toppings: each letter (between a & z) corresponds to a different topping, and the topping order is clockwise (the toppings form a circle around the outer edge of the pizza).

Write a program that, given a non-empty string less than 100 characters in length describing the sequence of toppings, return the maximum number of equal parts that can be cut from the pizza without leaving any leftovers.

Input format:

The input file contains `N`, where `1 <= N <= 100`, followed by `N` strings of lengths between `1` and `100`.

Output format:

Each line will contain the number of equal parts that can be cut without leaving any leftovers.

Sample Test Case 1

```
2
abcabcabcabc
abccbaabccba
```

Sample Output 1

```
4
2
```

Problem C

Study Groups

FCIT wants to pair up the best students of FCIT in a group so that they can share their areas of expertise with others. Each student may be well-versed in a number of academic subjects or topics. Given a list of topics known by each student, presented as binary strings, your task is to determine the maximum number of topics a 2-student team can know.

Each subject has a column in the binary string, and a '1' means the subject is known while '0' means it is not. Also, determine the number of groups that know the maximum number of topics. Think of it like building the ultimate study group. You want to find the pair of students who together know the most number of answers to the questions. In the end, return an integer array with two elements.

Return an integer array with two elements. The first is the maximum number of subjects known, and the second is the number of groups that know that number of subjects. It is time to put your coding skills to the test and find the ultimate study group at FCIT!

Example

```
Students = 3
Knowledge = ['01111', '10101', '00010']
```

The attendee data is in alignment for clarity below:

```
01111
10101
00010
```

Total possible combinations of teams are:

Students	Topics
(1,2)	[1,2,3,4,5]
(1,3)	[1,3,4,5]
(2,3)	[1,2,3,4]

In this case, the first group will know all 5 subjects. They are the only team that can be created that knows that many subjects, so it is returned.

Input format:

The first line contains two space-separated integers 'n' and 'm', where 'n' is the number of students and 'm' is the number of topics. The next 'n' lines contains the binary string of length 'm' with information about 'm' subjects.

Output format:

Print the maximum topics and the number of students that know these topics.

Sample Test Case 1

```
4 5
10101
11100
11010
00101
```

Sample Output 1

```
5
2
```

Sample Test Case 2

```
6 5
11101
10101
11001
10111
10000
01110
```

Sample Output 2

```
5
6
```

Problem D

Assalam o Alaikum FCIT

In FCIT Old Campus, some students enter from the front gate and some enter from PUCAD's gate. Students that enter from one gate need to stop and say Assalam o Alaikum to the students entering from the other gate. This takes a lot of time and it is highly inefficient so this is why the principal has hired you to write a program that will calculate the inefficiency of this procedure.

Every time the Students pass each other in the parking area, they must stop and greet one another before continuing on their way. Each greeting takes 1 minute since each greeting is 30 seconds long (Students also discuss course work). You believe that eliminating the small talk will remove the fun of students (which is what we want). But first, you must demonstrate to the Principal how serious the matter is.

Write a program that counts how many greetings are exchanged during a typical walk along a hallway. The hall is represented by a string. For example:

"~~>~><~>~~>"

Each string will contain three different types of characters: '>', a student walking to the right; '<', a student walking to the left; and '~', an empty space. Every student walks at the same speed either to the right or to the left, according to his or her direction. Whenever two students cross, each of them greets the other. They then continue walking until they reach the end, finally leaving the hallway. In the above example, they greeted each other 10 times.

Write a program which takes a string representing students walking in the parking and returns the number of times the students greeted each other. The string will contain at least 1 and at most 100 characters, each one of ~, >, or <.

Input format:

The input file contains N, where 1 <= N <= 100, followed by N strings of lengths between 1 and 100.

Output format:

Each line will contain the number of greeting(s) for the students

Sample Test Case 1

```
2
>~~~~<
<<>><
```

Sample Output 1

```
2
4
```

```
n = int(input())
for i in range(n):
    s = input()
```

AD's gate. Students
n the other gate.
ite a program

Problem E

Boxes Trap

A company manufactures LED bulbs. The company delivers for any quantity greater than two and less than one million. The company has large, medium, and small boxes. The size of a large box is 100 units, the size of a medium box is 20 units and the size of a small box is 5 units.

The company policy is to keep each box at least more than half-full (the exception is a small box). Otherwise, they have to use smaller size boxes. The company wants to quickly fill their orders. They employed you to write a program to find the number of each type of box to pack the order.

You have to read the number of bulbs in each order and print the number of each box (only if the box count is non-zero.

Input:

A single number showing a number of bulbs, where 2 <= number of bulbs < 1000000.

Output:

Minimum one line and maximum of three lines having a count of each type of box. Do not write information for box type in case of zero counts.

Sample:

Input

22

Output:

```
medium box = 1
small box = 1
```

Input

678

Output:

```
large box = 7
```

Input

432

Output:

```
large box = 4
medium box = 2
```

Input

422

Output:

```
large box = 4
medium box = 1
small box = 1
```

Problem F

The Goldmine Problem

Given a gold mine of M*N dimensions. Each field in this mine contains a positive integer, which is the amount of gold in tons. Initially, the miner is in the first column but can be in any row. The miner can move only (right-right, up-right-down) that is from a given cell, the miner can move to the cell diagonally up towards the right or diagonally down towards the right. Find out the maximum amount of gold that miner can collect.

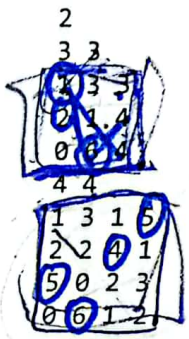
Input

The first line of the input gives the number of test cases, T. T test cases follow. Each case consists of one line containing two integers, M and N, and then M lines and each line contains N numbers.

Output

For each test case, output one line with Case #x: y, where x is the test case number (starting from 1) and y is the maximum Gold that miner can collect.

Sample Test Case 1



Sample Output 1

```
Case #1: 12

Case#2: 16
```

In Sample Case #1, Start from 2, and then it will go to 6 and in the end 4. Therefore, the max possible gold that miner can collect is 12.