Computer Programming
Assignment 5
Deadline: Friday 18th Nov., 5 PM

Note: You are allowed to upload/use the Online Judge for only three times per question. All further attempts are penalized. A 4th attempt will fetch you only 50%, and a 5th attempt will fetch you only 25% of the marks.

Q1:

Given a string s and a non negative integer k . Rotate the string circularly right such that the vowels remain in their original position and only the consonants move by a factor of k. String size will be less than 1000, k will be less than 10^6.

Input:

2
bceod 2
programming 13

Output:

cdeob
rmomnagpirg

Q2:

Given a non-empty string composed of digits only, we may cut this string into sub-strings (but maintaining their original order) if, for every sub-string but the last one, the sum of the digits in a sub-string is less than or equal to the sum of the digits in the sub-string immediately on its right. Needless to say, each digit will be in exactly one sub-string.

For example, the string 635 can only be grouped in one sub-group [635] or in two sub-groups as follows: [6-35] (since 6 < 8.) Another example is the string 1117 which can be grouped in one sub-group [1117] or as in the
following: [1-117], [1-1-17], [1-11-7], [1-1-1-7], [11-17],and [111-7] but not any more, hence the total number of possibilities is 7.

Write a program that computes the total number of possibilities of such groupings for a given string of digits.

Input

Your program will be tested on a number of test cases. Each test case is specified on a separate line. Each line contains a single string no longer than 25, and is made of decimal digits only.

The end of the test cases is identified by a line made of the word "bye" (without the quotes.) Such line is not part of the test cases.

Output

For each test case, write the result using the following format:
k. n
where k is the test case number (starting at 1,) and n is the result of this test case.

Sample

input
635
1117
9876
bye

output
1. 2
2. 7
3. 2

Q3:

Given a string, suppose s . Find the length of the maximum length palindrome which is a substring of the given string s. String size will be less than 1024.

Input:
2
aabb
malayalam

Output:
2
9

Q4:

REMOVING NESTED COMMENTS:
Given a C-code as the input, remove comments of the form '/*' from it. The main problem is that these comments are nested for eg: " /* Is this /* a nested */ comment ??

*/ “.

INPUT

C-Code is given as an input. You have to scan your input character by character till EOF.

OUTPUT

The output should be the same C-Code free from all the nested comments. Don’t remove comments if they are inside “ “.

Input:

```
#include<stdio.h>
int main(){
/* My Code /* Starts */ Here */
printf(" Hello /* take care /* of this */ thing too!! */ ");
return 0;
}
```

Output:

```
#include<stdio.h>
int main(){

printf(" Hello /* take care /* of this */ thing too!! */ ");
return 0;
}
```

Q5:

A simple game consists of a grid of **R**x**C** buttons. Each button will be either lighted, or unlighted. Whenever you push a button, the state of that button, and its (up to) four neighbors will toggle -- lighted buttons will become unlighted and unlighted buttons will become lighted. Note that the neighbors do not 'wrap' and thus a corner button has only two neighbors, while an edge buttons has three.

In this problem you will be given an initial configuration of the buttons. Your task is to push the right buttons so that, when you are done, all of the lights are turned on. If there are multiple ways to do this, you should determine the minimum number of buttons pushes that it can be done in.

Input

You will first read an integer **N** the number of test cases. For each test case, you will read two integers **R** and **C**. This will be followed by **R** whitespace-separated tokens, each containing **C** characters. A 'X' indicates a lighted button, while a '.' indicates an unlighted button.

Constraints

- **N** = 20

- 1 ≤ **R**,**C** ≤ 18

Output

For each test case you should output the minimum number of button presses required to turn on all the lights. If there is no way to do this, you should output -1.

input:

```
5

5 6
XXXXXX
XXX.X.
XXXXXX
X.XXXX
XXXXX.
1 13   ..XXXXXXX.X..
11 6
XXXXXX
XXXXXX
XXXXXX
XXXXXX
XXXXXX
XXXXXX
XXXXXX   .X.XXX
XXXX.X
XXXXXX
XXX.XX
10 13   ..XX...X.X.X.
XX..X..X.....   .X...........
X........X...   .....XX..X.X.   .X..XX.......   .X.....X.X...   .X....X......   ......XX...X.   ..X....X.....
9 3   ...   ...   ...   ...   ...   ..X   ...   ...   ...
```

output:

```
14
7
27
65
11
```

Q6:

Run-length encoding of a number replaces a run of digits (that is, a sequence of consecutive equivalent digits) with the number of digits followed by the digit itself. For example, 44455 would become 3425 (three fours, two fives). Note that run-length

encoding does not necessarily shorten the length of the data: 11 becomes 21, and 42 becomes 1412. If a number has more than nine consecutive digits of the same type, the encoding is done greedily: each run grabs as many digits as it can, so 111111111111111 is encoded as 9161.
Implement an integer arithmetic calculator that takes operands and gives results in run-length format. You should support addition, subtraction, multiplication, and division. You won't have to divide by zero or deal with negative numbers.

Input/Output

The input will consist of several test cases, one per line. For each test case, compute the run-length mathematics expression and output the original expression and the result, as shown in the examples. The (decimal) representation of all operands and results will fit in **signed 64-bit integers.**

Example

Input:
11 + 11
988726 - 978625
12 * 41
1124 / 1112
13 * 33
15 / 16

Output:
11 + 11 = 12
988726 - 978625 = 919111
12 * 41 = 42
1124 / 1112 = 1112
13 * 33 = 39
15 / 16 = 10