code jam

cout << "hello, world!" << endl;

Practice Mode

Qualification Round 2016

Contest Analysis

Questions asked 1

- Submissions

Counting Sheep

7pt	Not attempted **26558/29356 users** correct (90%)
8pt	Not attempted **25729/26216 users** correct (98%)

Revenge of the Pancakes

10pt	Not attempted **22527/23686 users** correct (95%)
10pt	Not attempted **21383/22147 users** correct (97%)

Coin Jam

10pt	Not attempted **13361/15342 users** correct (87%)
20pt	Not attempted **6297/9111 users** correct (69%)

Fractiles

10pt	Not attempted **8250/9708 users** correct (85%)
25pt	Not attempted **2356/4955 users** correct (48%)

- Top Scores

Lewin	100
Endagorion	100
xiaowuc1	100
xyz111	100
HellKitsune123	100
h4tguy	100
ivan.popelyshev	100

Problem B. Revenge of the Pancakes

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input
10 points

Download B-small-practice.in

your output file:

Browse... No file selected.

source file(s): not needed for the practice contest

Submit file Hide

Large input
10 points

Download B-large-practice.in

your output file:

Browse... No file selected.

source file(s): not needed for the practice contest

Submit file Hide

Problem

The Infinite House of Pancakes has just introduced a new kind of pancake! It has a happy face made of chocolate chips on one side (the "happy side"), and nothing on the other side (the "blank side").

You are the head waiter on duty, and the kitchen has just given you a stack of pancakes to serve to a customer. Like any good pancake server, you have X-ray pancake vision, and you can see whether each pancake in the stack has the happy side up or the blank side up. You think the customer will be happiest if every pancake is happy side up when you serve them.

You know the following maneuver: carefully lift up some number of pancakes (possibly all of them) from the top of the stack, flip that entire group over, and then put the group back down on top of any pancakes that you did not lift up. When flipping a group of pancakes, you flip the entire group in one motion; you do *not* individually flip each pancake. Formally: if we number the pancakes 1, 2, ..., N from top to bottom, you choose the top i pancakes to flip. Then, after the flip, the stack is i, i-1, ..., 2, 1, i+1, i+2, ..., N. Pancakes 1, 2, ..., i now have the opposite side up, whereas pancakes i+1, i+2, ..., N have the same side up that they had up before.

burunduk3	100
Nicolas16	100
ctunoku	100

For example, let's denote the happy side as + and the blank side as `-`. Suppose that the stack, starting from the top, is `--+-`. One valid way to execute the maneuver would be to pick up the top three, flip the entire group, and put them back down on the remaining fourth pancake (which would stay where it is and remain unchanged). The new state of the stack would then be `-++-`. The other valid ways would be to pick up and flip the top one, the top two, or all four. It would not be valid to choose and flip the middle two or the bottom one, for example; you can only take some number off the top.

You will not serve the customer until every pancake is happy side up, but you don't want the pancakes to get cold, so you have to act fast! What is the smallest number of times you will need to execute the maneuver to get all the pancakes happy side up, if you make optimal choices?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each consists of one line with a string **S**, each character of which is either + (which represents a pancake that is initially happy side up) or `-` (which represents a pancake that is initially blank side up). The string, when read left to right, represents the stack when viewed from top to bottom.

Output

For each test case, output one line containing `Case #x: y`, where x is the test case number (starting from 1) and y is the minimum number of times you will need to execute the maneuver to get all the pancakes happy side up.

Limits

1 ≤ **T** ≤ 100.
Every character in **S** is either + or `-`.

Small dataset

1 ≤ length of **S** ≤ 10.

Large dataset

1 ≤ length of **S** ≤ 100.

Sample

Input	Output
5 - -+ +- +++ --+-	Case #1: 1 Case #2: 1 Case #3: 2 Case #4: 0 Case #5: 3

In Case #1, you only need to execute the maneuver once,

flipping the first (and only) pancake.

In Case #2, you only need to execute the maneuver once, flipping only the first pancake.

In Case #3, you must execute the maneuver twice. One optimal solution is to flip only the first pancake, changing the stack to `--`, and then flip both pancakes, changing the stack to ++. Notice that you cannot just flip the bottom pancake individually to get a one-move solution; every time you execute the maneuver, you must select a stack starting from the top.

In Case #4, all of the pancakes are already happy side up, so there is no need to do anything.

In Case #5, one valid solution is to first flip the entire stack of pancakes to get `+-++`, then flip the top pancake to get `--++`, then finally flip the top two pancakes to get ++++.