Google Code Jam '18 Qualification Round Problem A - Saving The Universe Again

Time limit: 20.0s **Memory limit:** 1G

An alien robot is threatening the universe, using a beam that will destroy all algorithms knowledge. We have to stop it!

Fortunately, we understand how the robot works. It starts off with a beam with a strength of 1, and it will run a program that is a series of instructions, which will be executed one at a time, in left to right order. Each instruction is of one of the following two types:

- `C` (for "charge"): Double the beam's strength.
- `S` (for "shoot"): Shoot the beam, doing damage equal to the beam's current strength.

For example, if the robot's program is `SCCSSC`, the robot will do the following when the program runs:

- Shoot the beam, doing 1 damage.
- Charge the beam, doubling the beam's strength to 2.
- Charge the beam, doubling the beam's strength to 4.
- Shoot the beam, doing 4 damage.
- Shoot the beam, doing 4 damage.
- Charge the beam, increasing the beam's strength to 8.

In that case, the program would do a total of 9 damage.

The universe's top algorithmists have developed a shield that can withstand a maximum total of D damage. But the robot's current program might do more damage than that when it runs.

The President of the Universe has volunteered to fly into space to hack the robot's program before the robot runs it. The only way the President can hack (without the robot noticing) is by swapping two adjacent instructions. For example, the President could hack the above program once by swapping the third and fourth instructions to make it `SCSCSC`. This would reduce the total damage to 7. Then, for example, the president could hack the program again to make it `SCSSCC`, reducing the damage to 5, and so on.

To prevent the robot from getting too suspicious, the President does not want to hack too many times. What is this smallest possible number of hacks which will ensure that the program does no more than D total damage, if it is possible to do so?

Input Specification

The first line of the input gives the number of test cases, T. T test cases follow. Each consists of one line containing an integer D and a string P: the maximum total damage our shield can withstand, and the robot's program.

Output Specification

For each test case, output one line containing `Case #x: y`, where x is the test case number (starting from 1) and y is either the minimum number of hacks needed to accomplish the goal, or `IMPOSSIBLE` if it is not possible.

Limits

$1 \leq T \leq 100$.

$1 \leq D \leq 10^9$.

$2 \leq \text{length of } P \leq 30$.

Every character in P is either `C` or `S`.

Time limit: 20 seconds per test set.

Memory limit: 1GB.

Test set 1

The robot's program contains either zero or one `C` characters.

Test set 2

No additional restrictions to the Limits section.

Sample Input

```
6
1 CS
2 CS
1 SS
6 SCCSSC
2 CC
3 CSCSS
```

Sample Output

```
Case #1: 1
Case #2: 0
Case #3: IMPOSSIBLE
Case #4: 2
Case #5: 0
Case #6: 5
```

Note that the last three sample cases would not appear in test set 1.

In Sample Case #1, the President can swap the two instructions to reduce the total damage to 1, which the shield can withstand.

In Sample Case #2, the President does not need to hack the program at all, since the shield can already withstand the 2 total damage it will cause.

In Sample Case #3, the program will do more damage than the shield can withstand, and hacking will do nothing to change this. The universe is doomed.

Sample Case #4 uses the program described in the problem statement. The statement demonstrates one way to reduce the total damage to 5 using two hacks. It is not possible to reduce the damage to 6 or less by using only one hack; remember that the President can only swap adjacent instructions.

In Sample Case #5, the robot will never shoot, and so it will never do any damage. No hacking is required.

In Sample Case #6, five hacks are required. Notice that even if two hacks swap the instructions at the same two positions, they still count as separate hacks.