Saving the Universe Again

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 Format
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 Format
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.
Constraints

- 1 ≤ **T** ≤ 100
- 1 ≤ **D** ≤ 10^9
- 2 ≤ length of **P** ≤ 30
- Every character in **P** is either C or S.
- Time limit: 20 seconds per test set.
- Memory limit: 1GB.

Sample Input	Sample Output
6 1 CS 2 CS 1 SS 6 SCCSSC 2 CC 3 CSCSS	Case #1: 1 Case #2: 0 Case #3: IMPOSSIBLE Case #4: 2 Case #5: 0 Case #6: 5