Problem 1: Fairly Filtering Freshman [HackerRank]

By Sacheth Sathyanarayanan

You are a Machine Learning specialist with an expertise in Data Visualization and Aggregation (i.e. you make pretty graphs in Excel) by day, Princeton Tower Club bouncer by night. It’s Thursday night and your club is going PUID; consequently, you see a long queue of students waiting outside your club. Some of these students are freshman, while others are not.

Unfortunately, the Cannon (also PUID) bouncer called in sick, so every student who wanted to go to either Tower or Cannon is outside your club. You have to assign these students to either Tower or Cannon. Being the voice of fairness, you decide that both clubs should get an equal number of students, and furthermore, each club should also get an equal number of freshman (since, of course, both clubs want to minimize this number). Being lazy, you want to minimize the number of contiguous blocks of students going to the same club (assume that the first student gets assigned to Tower). Being a nerd, you decide to write a program to do this for you.

Here is a more formal description: your program takes as input (from standard input) a string where each character is either an ’F’ (for freshman) or an ’N’ (for non-freshman). This string represents the queue right outside Tower. Your task is to dice this line up into as few segments as possible, where all students in each segment are assigned to the same club. Remember: this is under the constraint that each of the two clubs must get the same number of freshmen, and each of the two clubs must get the same number of non-freshmen. Your program must output the index (start counting from zero) of the first students in each segment as well as which club they’re assigned to. If it is impossible to make such an assignment, print -1. Also note that it may be possible for such an assignment not to be unique; in such cases, output the assignment where the number of of students in first segment is minimized. Break any further ties by minimizing the number of students in the second segment, then in the third segment, and so on.

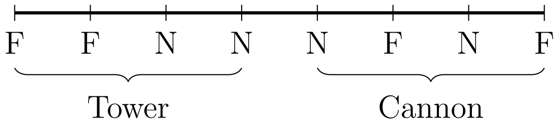
Input

The first, and only line, consists of a string of length n where each character is either an ‘F’ (for freshman) or ‘N’ (for non-freshman).

Output

The number of lines in your output should equal the number k of segments you wish to dice the line into. The m^{th} line (for $0 \leq m \leq k-1$) contains the index of the first student in the m^{th} segment followed by a space and either the character ‘T’ or ‘C.’ Here, ‘T’ means that you assign all students in the m^{th} segment to Tower while ‘C’ means that you assign them to Cannon.

Example 0



Input:

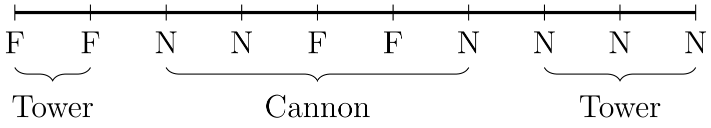
```
FFNNNFNF
```

Output:

```
0 T
4 C
```

Explanation: The minimum number of segments we can make here is two, where the first four students go to Tower and the last four students go to Cannon.

Example 1



Input:

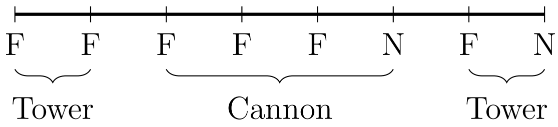
```
FFNNFFNNNN
```

Output:

```
0 T
2 C
7 T
```

Explanation: The minimum number of segments we can make here is three, where the first two students go to Tower, the next five students go to Cannon and the last three students go to Tower.

Example 2



Input:

```
FFFFFNFN
```

Output:

```
0 T
2 C
6 T
```

Explanation: The minimum number of segments we can make here is three, where the first two students go to Tower, the next four students go to Cannon and the last two students go to Tower. Note that there is another assignment where the first three students go to Tower, the next four students go to Cannon and the last student goes to Tower. We reject this assignment, however, since it has a greater number of students in the first assignment than what is presented in the solution.