

Given a line of text, find the longest almost-palindrome substring. A string S is almost-palindrome if

1. S begins and ends with a letter, and
2. $a(S)$ and $b(S)$ have at most $2k$ positions with different characters

Here $a(S)$ is the string after removing all non-letter characters and converting all the letters to lowercase, $b(S)$ is the reversed string of $a(S)$.

For example, when $k = 1$, “Race cat” is almost-palindrome, because $a(S)$ =“racecat” and $b(S)$ =“tacecar” differ at exactly 2 positions.

Input

There will be at most 25 test cases. Each test case contains two lines. The first line is k ($0 \leq k \leq 200$). The second line contains a string with at least one letter and at most 1,000 characters (excluding the newline character). The string will only contain letters, spaces and other printable characters like (‘,’ or ‘.’ etc) and *will not start with a whitespace*.

Output

For each test case, print the length of the longest almost-palindrome substring and its starting position (starting from 1). If there is more than one such string, print the smallest starting position.

Sample Input

```
1
Wow, it is a Race cat!
0
abcdefg
0
Kitty: Madam, I'm adam.
```

Sample Output

```
Case 1: 8 3
Case 2: 1 1
Case 3: 15 8
```