PROBLEM 2

Subbookkeeper

Time and memory limits: 1 second, 1 GB

Rebecca loves words, and has devised a system to calculate which words are her favourite. She gives each word a *score*, which is the number of times that two adjacent letters are the same:

- `REBECCA` has a score of 1, because it has adjacent `C`s.
- `ANNABELLA` has a score of 2, because it has adjacent `N`s and adjacent `L`s.
- `HELLLLLO` has a score of 4, because `LLLLL` has four sets of adjacent `L`s.
- `HAHAHA` has a score of 0, because it has no adjacent letters that are the same.

Rebecca has recently started a job as the subbookkeeper of the Woolloomooloo library, where she ranks 2nd only to the main bookkeeper Annabella. Part of her job involves inspecting books to see if any letters have rubbed off the pages. She must then write in the missing letters. Rebecca has decided to choose the replacement letters so that the score of the word is as large as possible. Rebecca's words are simply groups of letters. They don't have to be real words from the dictionary.

For example, imagine that she found the word `RE?EL`, where `?` represents a missing letter. If she chose `P`, then it would make the word `REPEL` with a score of 0. However, if she instead chose `E`, this would make `REEEL` with a score of 2.

Rebecca has found a word with exactly one missing letter. What is the largest score she can achieve by strategically choosing the replacement letter?

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

- $3 \leq N \leq 200\,000$, where N is the number of letters in the word.
- The word consists of uppercase letters from `A` to `Z`, with exactly one `?`.

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (30 marks), the `?` is the 1st letter in the word.
- For Subtask 2 (40 marks), the `?` is not the 1st nor the last letter in the word.
- For Subtask 3 (30 marks), no special rules apply.

Input

We strongly recommend using the solution templates (which you can find at the *Attachments* section if you scroll down on ORAC) to help you with input and output.

Your program must read input in a specific format:

- The 1st line contains the integer N, which is the number of letters in the word.
- The 2nd line contains the word: exactly `N` characters, where one is `?` and the others are uppercase letters.

Output

Your program must print a single integer: the largest score Rebecca can achieve by strategically choosing the replacement letter.

Sample input 1

```
5
RE?EL
```

Sample input 2

```
8
?ELLLLLO
```

Sample input 3

```
13
SUBBOO?KEEPER
```

Sample output 1

```
2
```

Sample output 2

```
5
```

Sample output 3

```
4
```

Explanation

- In the 1st sample case, the best word Rebecca can make is `REEEL` with a score of 2.
- In the 2nd sample case, the best word she can make is `EELLLLLO` with a score of 5 (1 from the `E`s and 4 from the `L`s).
- In the 3rd sample case, she can make either `SUBBOOKKEEPER` or `SUBBOOOKEEPER`, which both have a score of 4.