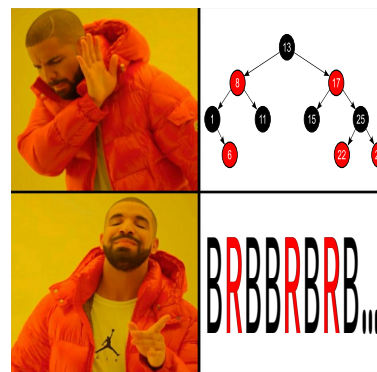


I Intricate Idioms

Time limit: 3s

Ilya is a first-year Computer Science student, but more importantly, he is a language enthusiast. He has a very large collection of idioms that he spent years procuring. However, this collection has started to get out of hand, so Ilya is now looking to organize it. Recently, during the Algorithms & Data Structures lecture, he heard about some binary trees which are coloured in two colours, red and black. He left the lecture quite confused, but very interested about the topic, looking to find some potential uses for them while organizing his collection. Later, while trying to understand the red-black property, he converted all the trees to strings. However, something went wrong with his translation and he is now looking for your help to tell him how many steps he needs to convert them correctly.



Ilya after converting the trees to strings.
Internet meme: “Drakeposting”, fair use
Tree: CC BY-SA 4.0 by Nomen4Omen
on Wikimedia Commons

You are given n strings consisting of the characters ‘R’ (Red) and ‘B’ (Black). You need to make sure that the red-black property is satisfied for all strings, in as few moves as possible. In one move, you can replace one character from ‘R’ to a ‘B’ (but not vice versa). Ultimately, the following properties need to be satisfied:

- The leftmost character of each string is a ‘B’.
- In each string, there can be no two ‘R’ characters adjacent to one another.
- The number of ‘B’ characters in each string needs to be the same as the number of ‘B’ characters in every other string.

What is the minimum number of moves to make the strings satisfy all properties?

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 2 \cdot 10^5$), the number of strings.
- n lines, each with a string s ($1 \leq |s| \leq 2 \cdot 10^5$), each character being either ‘B’ or ‘R’.

The sum of lengths over all strings is at most $2 \cdot 10^5$.

Output

If it is possible to make the strings satisfy all properties, output the minimum number of moves needed to do so. Otherwise, output “impossible”.

Sample Input 1

```
1
RBBRRBBBR
```

Sample Output 1

```
2
```

Sample Input 2

```
3
BRBRR
BRB
RBB
```

Sample Output 2

```
3
```

Sample Input 3

```
2
BBB
RR
```

Sample Output 3

```
impossible
```