

DMPG '17 S6 - Brackets, Braces, and Boxes

Molly was experimenting with bracket strings, when she made a mistake and accidentally changed some of the characters. Now, she needs your help to fix her bracket sequence.

To begin, here are Molly's definitions:

- A bracket string consists of the following six symbols: `()[]{}.` The characters `{[` are considered left brackets, and `)]}` are considered right brackets.
- There are three distinct matching pairs: `()`, `[]`, and `{}`. Within each pair, the left bracket must be paired with the right bracket.
- A string is considered **valid** if every left bracket can be paired with a distinct right bracket such that the left bracket appears to the left of the right bracket, and the string inside of the matching pair is either empty or valid. For example, valid sequences could be `()`, `{[][(())]}`, or `()[]{}{[(())]}` whereas invalid sequences would be `({})`, `((([)])` and `}{}.`

Currently, Molly has a bracket string (either valid or invalid), and would like you to change **exactly** K symbols at distinct positions, and create a valid string. However, **you are not allowed to change a left bracket to a right bracket, or a right bracket to a left one.** If it is impossible to create a valid bracket string, output `impossible` instead.

Subtasks

Subtask 1 [20%]

$$N, K \leq 20$$

Subtask 2 [30%]

$$K \leq 1$$

Subtask 3 [50%]

No additional constraints.

Input Specification

On the first line, there will be the integers N and K . On the second line will be the original string of length N .

For all cases, $2 \leq N \leq 200\,000$, and N is even; and $0 \leq K \leq N$.

Output Specification

On a single line, print either a valid bracket string of length N with K characters different from the input string, or `impossible`.

Sample Input 1

```
2 2  
( )
```

Sample Output 1

```
[ ]
```

Sample Input 2

```
4 2  
( { } )
```

Sample Output 2

```
( { } )
```

Sample Input 3

```
6 2  
( ( ( [ ] ] ) ) )
```

Sample Output 3

```
impossible
```