https://course.acciojob.com/idle?question=8eb53aa6-bb90-4a5c-8995-b4a8383dc4ee

- EASY
- Max Score: 30 Points
-

Remove Outermost Parentheses

A valid parentheses string is either empty `""`, `"(" + A + ")"`, or `A + B`, where `A` and `B` are valid parentheses strings, and `+` represents string concatenation.

For example, `""`, `"()"`, `"(())()"`, and `"(()(()))"` are all valid parentheses strings.

A valid parentheses string `s` is primitive if it is nonempty, and there does not exist a way to split it into s = A + B, with A and B nonempty valid parentheses strings.

Given a valid parentheses string s, consider its primitive decomposition: s = P1 + P2 + ... + Pk, where Pi are primitive valid parentheses strings.

print s after removing the outermost parentheses of every primitive string in the primitive decomposition of s.

Input Format

The first line contains a single integer `n`(length of string)

Second line conatins the string `s` of size `n`

Output Format

print the modified string

Example 1

Input

```
10
```

```
(()())(())
```

Output

```
()()()
```

Explanation

The input string is "(()())(())", with primitive decomposition "(()())" + "(())".

After removing outer parentheses of each part, this is "()()" + "()" = "()()()".

Example 2

Input

```
18
(()())(())(()(()))
```

Output

```
()()()()(())
```

Constraints

1 <= s.length <= 10^5

s[i] is either '(' or ')'.

s is a valid parentheses string.

Topic Tags

- Strings
- Stacks

My code

```
// in java
import java.util.*;
import java.lang.*;
import java.io.*;

public class Main
{
        public static void main (String[] args) throws java.lang.Exception
        {
                //your code here
      Scanner s=new Scanner(System.in);
      int n=s.nextInt();
      String str=s.next();
      Stack<Character> stk=new Stack<>();
      for(int i=0;i<str.length();i++)
        {  char ch=str.charAt(i);
         // if(ch=='(')  stk.push(ch);
        //  else { char c=stk.pop();
     //    if((stk.empty())==false) { System.out.print(c+""+ch);}}
if(ch=='(')
{
  if((stk.empty())==false) System.out.print(ch);
  stk.push(ch);
}
         else
{char c=stk.pop();
 if((stk.empty())==false) { System.out.print(ch);}}
```

```
        }



        }
    }
```