

# Remove Invalid Parenthesis

---

1. You are given a string, which represents an expression having only opening and closing parenthesis.
2. You have to remove minimum number of parenthesis to make the given expression valid.
3. If there are multiple answers, you have to print all of them.

Note -> Check out the question video and write the recursive code as it is intended without changing signature. The judge can't force you but intends you to teach a concept.

## Input Format

A string containing only opening and closing parenthesis

## Output Format

Print all the Valid expressions.

Check the sample output and question video.

## Constraints

$1 \leq \text{length of string} \leq 20$

## Sample Input

()())

## Sample Output

((()))

()()

```
def invalidParenthesis(string):
    minRemoval = getMin(string)
    ans = set()
    helper(string, ans, minRemoval)
    return ans

def helper(string, ans, x):
    if x == 0:
        if string not in ans and getMin(string) == 0:
            ans.add(string)
        return
```

```
for i in range(len(string)):
    left = string[:i]
    right = string[i + 1:]
    helper(left + right, ans, x - 1)
```

```
def getMin(string):
    stack = []
    for el in string:
        if el == '(':
            stack.append(el)
        else:
            if len(stack) == 0:
                stack.append(el)
            else:
                if stack[-1] == '(':
                    stack.pop()
                else:
                    stack.append(el)
    return len(stack)
```

```
print(invalidParenthesis('()())()'))
```