

**Problem 5:** Check Balanced Parentheses. Given string str containing just the characters '(', ')', '{', '}', '[', and ']', check if the input string is valid and return true if the string is balanced otherwise return false.

**Note:** string str is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

```
def is_balanced_parentheses(string):
```

```
    stack = []
```

```
    opening_brackets = ['(', '[', '{']
```

```
    closing_brackets = [')', ']', '}']
```

```
    bracket_map = {')': '(', ']': '[', '}': '{'}
```

```
    for char in string:
```

```
        if char in opening_brackets:
```

```
            stack.append(char)
```

```
        elif char in closing_brackets:
```

```
            if len(stack) == 0:
```

```
                return False
```

```
            if stack[-1] == bracket_map[char]:
```

```
                stack.pop()
```

```
        else:
```

```
            return False
```

```
return len(stack) == 0
```

```
input_string = "([]{})"
```

```
print(input_string)
```

```
print(is_balanced_parentheses(input_string))
```

```
input_string = "([])"
```

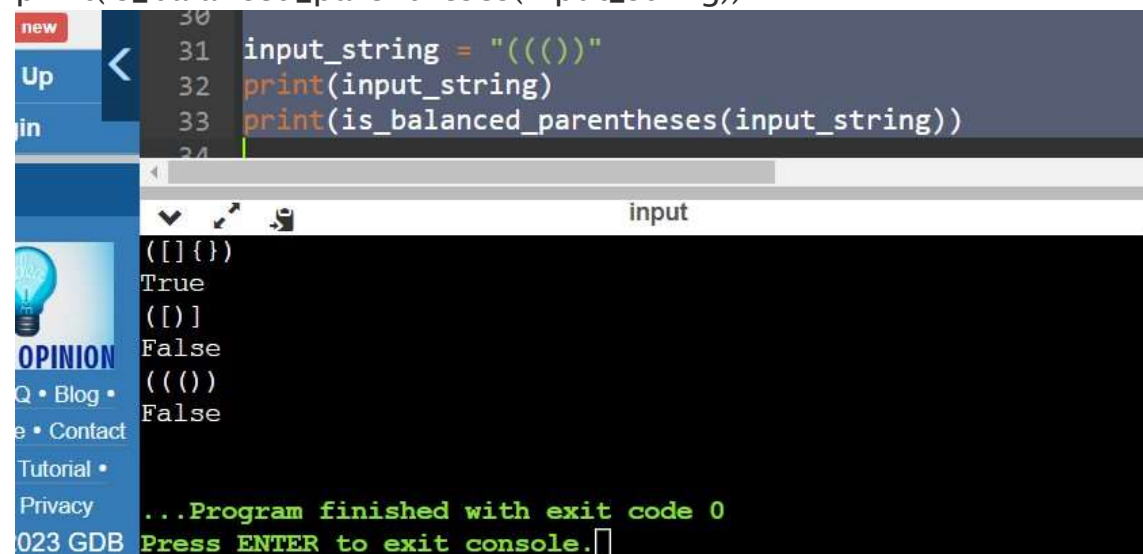
```
print(input_string)
```

```
print(is_balanced_parentheses(input_string))
```

```
input_string = "((()))"
```

```
print(input_string)
```

```
print(is_balanced_parentheses(input_string))
```



```
30
31 input_string = "((()))"
32 print(input_string)
33 print(is_balanced_parentheses(input_string))
34
```

input

```
([]{})
True
([])
False
((()))
False
```

...Program finished with exit code 0  
Press ENTER to exit console.