Challenge 8: Check Balanced Parentheses Using Stack

Check if parentheses in a given expression are balanced or not.

We'll cover the following

Problem Statement

In this problem, you have to implement the `is_balanced()` function which will take a string containing only curly `{}`, square `[]`, and round `()` parentheses. The function will tell us whether all the parentheses in the string are balanced or not.

For all the parentheses to be balanced, every opening parenthesis must have a closing one. The order in which they appear also matters. For example, `{[]}` is balanced, but `{[}]` is not.

Input

A string consisting solely of `(`, `)`, `{`, `}`, `[`, and `]`

Output

It returns `True` if the expression contains balanced parentheses and returns `False` otherwise.

Sample Input

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

Sample Output

```
True
```

Coding Exercise

Take a close look and design a step-by-step algorithm first before jumping on to the implementation. This problem is designed for your practice, so try to solve it on your own first. If you get stuck, you can always refer to the solution provided in the solution section. Good luck!

main.py

Stack.py

```
class MyStack:
```

```
    def __init__(self):
        self.stack_list = []
        self.stack_size = 0

    def is_empty(self):
        return self.stack_size == 0

    def peek(self):
        if self.is_empty():
            return None
        return self.stack_list[-1]

    def size(self):
        return self.stack_size

    def push(self, value):
        self.stack_size += 1
        self.stack_list.append(value)

    def pop(self):
        if self.is_empty():
            return None
        self.stack_size -= 1
        return self.stack_list.pop()
```