**Parentheses Check**

**Q:** Given two strings, write a method to decide if one is a permutation of the other.

**Test Cases:**

**Input:** “{}(([]))”

**Output:** “Valid”

**Input:** “{{{}}()”

**Output:** “Invalid”

**Input:** “{[()(())]}”

**Output:** “Valid”

**HINT:** If the opening parentheses doesn't have the matching closing parentheses than its an invalid string of parentheses.

**Algorithm:**

1) Create an empty stack

2) Read the character from the string one by one and if:

--> the character is the opening bracket ("(","{","[") then push it onto the stack

--> the character is closing bracket (")","}","]") than pop one character from the stack and if the corresponding closing bracket is popped from the stack than make the flag variable as true otherwise the parentheses are not matched.

3) Keep reading the character and goto step 2 till the end of the string

**Example:**

**Input:** {{}()

* Read the first opening curly bracket, push it on the stack.
* Read the second curly bracket, push it onto the stack.
* Read the next closing bracket and pop one bracket from the stack.(Matches)
* Read the next pair of opening and closing parentheses.
* At the end of the string, stack is not empty hence it's an Invalid string.