Spring 2025 MSML606 HW2

Instructions:

1. You are required to submit a Python file consisting of your code, a boilerplate Python template will be provided to write the code in. Test cases will be provided to you in csv files for both questions.
2. You are required to submit a report (pdf/doc) discussing your solutions and complexities.
3. Create a zip of both the files above and submit.

Rubrics:

Implementation	**Criteria**	**Marks**
Stack Implementation	Correctly implemented stack with required functions.	2
Valid Parentheses	Function correctly determines valid parentheses sequences.	2
Postfix Evaluation Function	Correctly evaluates a postfix expression using the common stack	2
Code Readability & Edge Cases	Code is well-structured and handles edge cases properly. (1.5 Marks for each problem)	3
Descriptive question	Description of usefulness of postfix expressions, and the process to convert infix to postfix	1
Report	Discuss your solutions and Big Oh complexities	2

Total: 12 Marks

[Problem 1] Valid Parentheses Using Stack

Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine *if* the input string is valid.

An input string 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.

3) Every close bracket has a corresponding open bracket of the same type.

Example 1:

Input: s = "()"

Output: true

Example 2:

Input: s = "()[]{}"

Output: true

[Problem 2] Postfix Calculator using stacks

1. Understand infix and postfix Expressions (no writeup or source code required to submit).
2. Briefly describe the usefulness of postfix expressions, and the process to convert infix to postfix.
3. Implement a postfix calculator: write your own code to create a calculator that evaluates postfix expressions using stack(s)

Write code to evaluate a postfix expression using stack and return the integer value. Use stack which is implemented above for this problem

Input -> an postfix expression string ex: "3 4 /"

Output -> integer value after evaluating the string ex: 1

Integers are positive and negative

Instructions:

DO NOT USE EVAL function for evaluating the expression.

The valid operators are '+', '-', '*', and '/'.