ECS 32B – Introduction to Data Structures
Homework 02

Due: October 16, 2020, 5:00pm PST

Important:

- Download the hw02-framework.py file from Canvas -> Assignments -> Homework02.
- Write your code (**in Python3**) in the designated positions in hw02-framework.py.
- The purpose of the homework is to practice using stacks. **The problems in all three functions must be solved using stacks.** The definition for Stack is included in hw02-framework.py. If you didn't use stacks, only half of the points will be given.
- Ignore the self parameter in each function. The autograder uses it to run test cases.
- To submit, **rename hw02-framework.py to hw02.py, compress it as a zip file**, and upload to Gradescope. Because of the way the autograder on Gradescope is configured, the file must be named hw02.py and compressed; otherwise, you will not get any points.
- You may upload as many times as you want before the deadline. Each time the autograder will let you know how many of the test cases you have passed/failed. To prevent people from manipulating the autogader, the content of the test cases are hidden.
- Please **do not copy others' answers**. Autograder can detect similar code. Also, for your own benefit, do not try to find solutions online.

Function 1: postfixEval (14 test cases, total 32 points)

Write a function named postfixEval such that given a postfix expression, evaluate it and return the result.

Assumptions:

- the operands are floats
- the possible operators are +, -, *, /
- the input expression is always a valid and none-empty postfix expression

Example: When input is ["2", "4", "7", "$*$", "$-$"], which is $2 - 4 * 7$ using infix notation, your function should -26.

Function 2: validParentheses (29 test cases, total 35 points)

Given a (possibly empty) string containing only alphabet letters and parentheses “(”, “)”, “{”, “}”, “[”, “]”, determine if the string is valid.

A string is valid if

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

Example: Given input “(ab){[]}”, your function should return True.

Function 3: reverseString (10 test cases, total 33 points)

Given a (possibly empty) string, reverse it.

Example: Given input “abcde”, your function should return “edcba”.