

CS313-Spring 2021  
Suyogya Rayamajhi  
Prof. Matthew Fried  
Homework 2

### **Problem 1**

Commenting the code was required for this question.

#### **Part a) Stack Linked List:**

For the “stack linked list” implementation firstly, the node is defined. The node contains data call info, and a pointer to another node. Function headers are listed before defining them below. stackTop and copyStack functions are kept private.

The Boolean function isFullStack always returns false because it is a linked list based stack.

InitializeStack serves as the delete function as it empties the stack.

The functions pop, top and push serve to remove the topmost element, return the topmost element and remove the topmost element, respectively.

The copy constructor makes a new stack, and then uses copyStack to transfer the contents of one stack to the new stack. The destructor uses initializeStack to delete all the nodes in the stack. Finally, the overloaded assignment operator checks that the expression is not saying the stack is equal to itself, as this would create redundancy. Then, it uses the copyStack function to copy otherStack into the assigned stack.

#### **Part b) Postfix.txt**

This program used a stack and filestream with four other functions in the main.

In main, it create a Boolean flag called expressionOk, a character called ch, and a new stack called stack. Inside the while loop, for each expression, it uses intializeStack to erase the stack, sets expressionOk to true, and writes the first character ch to the output file. Then, it calls

evaluateExpression, passing it the inputstream, outputstream, stack, character, and expressionOk flag. The evaluateOpr function first creates two doubles, op1 and op2. Then it checks if the stack is empty, which it shouldn't be.

If the stack is empty, the result is printed in the output file.

## **Problem 2**

We write a prefix stack & show how it functions.

We create a Boolean function to check if each character is a digit and a double function to evaluate the prefix. In the double function we create a double stack and have a for loop that iterates through the entire expression. We push the operand into the stack and check to see if we are working with a digit. If it is a digit we convert the expression to a digit by subtracting 0. If NOT a digit, i.e. an operator then we pop 2 elements from the stack and we perform a switch case to perform the appropriate operation. We returned it to the top of the stack and the main function acts as a driver code.

## **Problem 4**

We need to iterate through a stack, print all the elements and also replace an element in a given position of the stack. The tricky part here is the deletion of a random element because stack, unlike other Abstract Data Types such as deque, stacks do not allow random data access. Thus, we use a linked list to implement stack. We start by creating a stack class. We have method print and replace in it. The print method simply uses a while loop to print until the end of stack. It uses a Node\* temp that initially points to the head, then traverses through the stack performing temp = temp -> next. The replace method takes an int value which is the position of the stack element to replace and a string to replace it with. We create a counter that goes through

the list until we reach the desired position, then we replace that position's value with the value passed in the function.

### **Problem 6**

We use a stack to pass a string and reverse each word in the string. We read line by line from the input file and reverse each word in the string, using stack and write to an output file. The main function acts as the driver and a while loop is used to go through the input file. The reversed output is written to the output file and both input and output files are closed.

The reverse function goes through the entire string using a for loop and to group characters that form a word in the for loop it checks if the current character is a space, comma, period, quotation. While no such character is found, other characters are pushed into the stack. Each separation indicates a word. A function to reverse the string is called once this is determined and the function is also called if it is the last word in the string.

The wordreverse function uses the position of the current character in the string subtracted by the size of the word. The first character position is equal to the top of the stack and the letter count of the word is reset back to 0.