CSC 10A

Accelerated Introduction to Programming Logic

Lab #6

20 Points

Name: Rifat Khan

Stacks and Queues

The purpose of this assignment is to give you practice in evaluating code that uses stacks and queues, understanding the differences between push, pop, and peek, and understanding the difference between enqueue, dequeue, and peek. Also, you should understand what makes a stack or queue different from an array.

Requirements

Answer the questions below for credit. (Show your work if applicable)

Question #1 (2 points): Draw a stack diagram (on the right side of the code) with the values from the top to the bottom based on the following pseudo-code:

Create an empty stack that holds integers named “st”

st.push(12)

st.push(5)

st.push(14)

Print st.peek()

st.push(3)

st

3
14
5
12

Question #2 (2 points): Draw a queue diagram (on the right side of the code) with the values from the left side of the queue to the right based on the following pseudo-code:

Create empty queue that holds integers named “q”

q.enqueue(7)

Print q.peek()

q.enqueue(10)

q.enqueue(13)

Print q.dequeue()

q.enqueue(2)

q	2	13	10	7

Question #3 (2 points): Write pseudo-code on the right-hand side of the page (using the code in Question #1 as a guide) to create a stack of integers and populate it with the values seen in the stack diagram on the left side:

st

7
2
5

3
1

stack <int> st

st.push (1)

st.push (3)

st.push (5)

st.push (2)

st.push (7)

Question #4 (2 points): Write pseudo-code on the right-hand of the page (using the code in Question #2 as a guide) to create a queue of integers and populate it with the values seen in the queue diagram on the left-hand side:

q	9	4	2	7	8

queue <int> q

q.add (8)

q.add (7)

q.add (2)

q.add (4)

q.add (9)

Question #5 (2 points): Explain what makes the stack operations of push, pop, and peek different? (If you just wish to define each one of them, that is fine)

The operation of Push is to add a new integer into the stack or queue. Operation of Pop serves to take out an integer from any stack or queue. Lastly the operation of Peek is to keep an integer from the stack or queue without removing it.

Question #6 (2 points): Explain what makes the queue operations of enqueue, dequeue, and peek different? (If you just wish to define each one of them, that is fine)

Enqueue is used to add new integers into a queue. Dequeue is used to remove integers from the queue. Peek is used to take out an integer from the queue without removing it.

Question #7 (2 points): Explain what makes stacks and queues different from each other and what makes both of them different than arrays?

Stacks are based on LIFO (last-in, first-out) rule. Queues use FIFO (first-in, first-out) rule. The difference between stacks and queues and arrays is that stacks and queues only have a singular entry value and one exit value, but arrays have multiple entry values

Question #8 (4 points): Given a certain situation, circle which one of the data structures (stack, queue, or array) that you believe would be the most efficient to use (Research the uses of these data structures if needed!):

Example:

When you need to access any element quickly Stack / Queue / (Array)

Creating an undo / redo operation **Stack** / Queue / Array

When you need to store waiting processes in order Stack / **Queue** / Array

Keeping track of recursive function calls Stack / Queue / **Array**

When you don’t delete elements **Stack** / Queue / Array

Question #9 (2 points): Given a queue named “q” that looks as follows:

1	2	3	4	5

Write the pseudo-code that will place this into a stack called “st”

stack <int> st

st.push (5)

st.push (4)

st.push (3)

st.push (2)

st.push (1)

Explain how the conversion from a queue to a stack changed the composition of the data?

When you transition data from a queue to a stack, remove the first value in respect to the rules within the two data sets first. This is similar to stacking plates, taking them down and putting them back up, you notice that the box that at the beginning was at the top is now at the bottom. If you look deeper you’ll notice that the new stack is the inverse of the original one.