Homework #3

Data Structure (501324-3)

Fall 2023

Due: Thursday December 7, 2023, 11:59 pm via Blackboard

Required:

* Answer all the following questions.
* Implement and show the output of each linked list question.
* In each coding exercise, add a comment // your name // your id.
* NO homework will be accepted without this comment.
* Put everything in one PDF file and upload it.
* NO LATE homework or excuses will be accepted. I gave you enough time to work on this homework.

**Question 1:**

Write a C++ code to insert the following numbers into a linked list:

20, 9, 18, 8, 99

1. Write a user-defined function that prints this list.
2. Write a function that calculates and prints the average of these numbers.
3. Delete the node that carries the data 8 and display the linked list.

**Question 2:**

Write a C++ code that create a linked list with 12 random numbers between 1 and 100.

1. Write a user-defined function that prints this list.
2. Write a function that takes a number from the user and searches that entered number in linked list

* if you find the number in the list, move the node containing that number to the end of the linked list and print the new list.
* otherwise, tell the user that the number was not found.

**Question 3:**

Suppose we have a stack **S** and a queue **Q**. What are final values in the stack **S** and in the **Q** after the following operations? Show contents of both **S** and **Q** at each step indicated by the line.

Stack S;

Queue Q;

int x, y;

S.push(10);

S.push(20);

S.push(S.pop()+S.pop());

Q.enqueue(10);

Q.enqueue(20);

Q.enqueue(S.pop());

S.push(Q.dequeue()+Q.dequeue());

**Question 4:**

Suppose we have an integer-valued stack S and a queue Q. Draw the contents of both S and Q at each step indicated by the line. Be sure to identify which end is the top of S and the front of Q.

Stack S;

Queue Q;

S.push(3);

S.push(2);

S.push(1);

Q.enqueue(3);

Q.enqueue(2);

Q.enqueue(1);

int x = S.pop();

Q.enqueue(x);

x = Q.dequeue();

Q.enqueue(Q.dequeue());

S.push(Q.peek());

// peek() function reads the front of a queue without deleting it

**Question 5:**

What will be the content of queues Q1, Q2, and Stack S, after the following code segment?

while(!Q1.isEmpty())

{

x = Q1.Dequeue();

if (x == 1)

{

z = 0;

while(!S.isEmpty())

{

y = S.pop();

z = z + y;

}

Q2.Enqueue(z);

}

Else

S.push(x);

}

Stack S;

Queue Q1, Q2;

int x, y, z;

Q1.Enqueue(9);

Q1.Enqueue(6);

Q1.Enqueue(9);

Q1.Enqueue(1);

Q1.Enqueue(7);

Q1.Enqueue(5);

Q1.Enqueue(1);

Q1.Enqueue(2);

Q1.Enqueue(8);

**Question 6:**

Assume that you have a stack S, a queue Q, and the standard stack - queue operations: push, pop, enqueue and dequeue. Assume that print is a function that prints the value of its argument. Execute, in top-to-bottom order, the operations below and answer the following questions.

push(S, ‘T’);

enqueue(Q, ‘I’);

push(S,dequeue(Q));

enqueue(Q, ‘I’);

enqueue(Q, ‘G’);

print(dequeue(Q));

enqueue(Q, T);

push(S, ‘I’);

push(S, dequeue(Q));

print(pop(S));

enqueue(Q, pop(S));

push(S, ‘O’);

print(pop(S));

enqueue(Q, ‘O’);

print(dequeue(Q));

enqueue(Q, pop(S));

push(S, dequeue(Q));

print(pop(S));

print(pop(S));

**Question 7:**

1. Use stack to evaluate the following infix expressions

(3 \* 4 – (2 + 5)) \* 4 / 2

1. Use stack to evaluate the following postfix expressions

8 2 3 ^ / 2 3 \* + 5 1 \* -

1. Convert the following infix expression into postfix expression using stack.

(3 \* 4 – (2 + 5)) \* 4 / 2