TUORIAL 6

ECE532

PART A: LINKED LIST

Algorithm Workbench

1. Using the ListNode structure to create a node, write a function

    void printFirst(ListNode*ptr)

That points the value stored in the first node of a list passed to it as parameter. The function should print an error message and terminate the program if the list passed to it is empty.

2. Write a function

    void printSecond(ListNode*ptr)

That points the value stored in the second node of a list passed to it as parameter. The function should print an error message and terminate the program if the list passed to it has less than two nodes.

3. Write a function

    void printValue(ListNode*ptr)

That points the value stored in the last node of a nonempty list passed to it as parameter. The function should print an error message and terminate the program if the list passed to it is empty.

4. Write a function

    ListNode *removeFirst(ListNode*ptr)

That is passed a linked list as parameter, and returns the tail of the list: that is, it removes the first node and returns what is left. The function should deallocate the storage of the removed node. The function returns NULL if the list passed to it is empty.

5. Write a function

    ListNode *ListConcat(ListNode*list1, ListNode *list2)

That concatenates the items in list2 to the end of list1 and returns the resulting list.

Predict the Output

6. ListNode *p = new ListNode(56.4);

    p= new ListNode(34.2, p);

    cout << (*p). value << endl << p->value;

7. ListNode *p = new ListNode(56.4);

    p = new ListNode(34.2, p);

    ListNode * q = p->next;

```
cout << q->value;
```

8.
```
ListNode *p = new ListNode(56.4, new ListNode (31.5));
ListNode *q = p;
while (q-> next->next != NULL)
q = q ->next;
cout<< q->value;
```

PART B: STACK AND QUEUE

9. Suppose the following operations are performed on an empty queue:

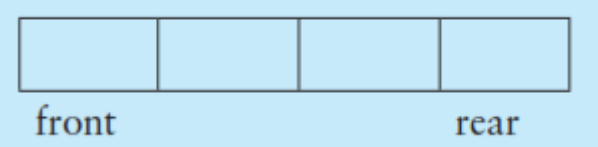
enqueue(5);

enqueue(7);

enqueue(9);

enqueue(12);

Insert numbers in the following diagram to show what will be stored in the static stack after the operations above have executed.



10. Suppose the following operations are performed on an empty queue:

enqueue(5);

enqueue(7);

dequeue();

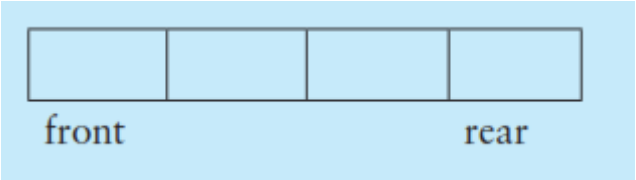
enqueue(9);

enqueue(12);

dequeue();

enqueue(10);

Insert numbers in the following diagram to show what will be stored in the static stack after the operations above have executed.



11. Convert the following infix expressions to postfix notations.

a. (A + B) * (C + D) - E

b. A - (B + C) * D + E / F

c. ((A + B) / (C - D) + E) * F - G

d. A + B * (C + D) - E / F * G + H

12. Write the equivalent infix expression for the following postfix expressions.

a. A B * C +

b. A B + C D - *

c. A B – C – D *