CMPE 242
Spring 2021
Hands-On Activity 3

Multiple Choice

1. One difference between a queue and a stack is:
   A. Queues require linked lists, but stacks do not.
   B. Stacks require linked lists, but queues do not.
   C. Queues use two ends of the structure; stacks use only one.
   D. Stacks use two ends of the structure, queues use only one.

2. If the characters 'D', 'C', 'B', 'A' are placed in a queue (in that order), and then removed one at a time, in what order will they be removed?
   A. ABCD
   B. ABDC
   C. DCAB
   D. DCBA

3. I have implemented the queue with a linked list, keeping track of a front node and a rear node with two reference variables. Which of these reference variables will change during an insertion into a NONEMPTY queue?

   A. Neither changes
   B. Only front changes.
   C. Only rear changes.
   D. An exception is caused

4. To simulate people waiting in a line, which data structure would you use?

   A. Vector
   B. Queue
   C Stack
   D. Set
   E. List

Coding Problem #1. Write a simple method that moves all of the contents of a linked list into a queue. Use the standard linked list and queue operations. After the move is complete, the first node in the linked list should be at front of the queue.

```
public class Node
{
     String item;
     Node next;
}


public class SLinkedList
{
     private Node head = null;


     // Doubly linked list methods
     …
}


// Move all of the contents of the linked list onto a queue
public void moveContents(SLinkedList list, Queue q)
{









}
```

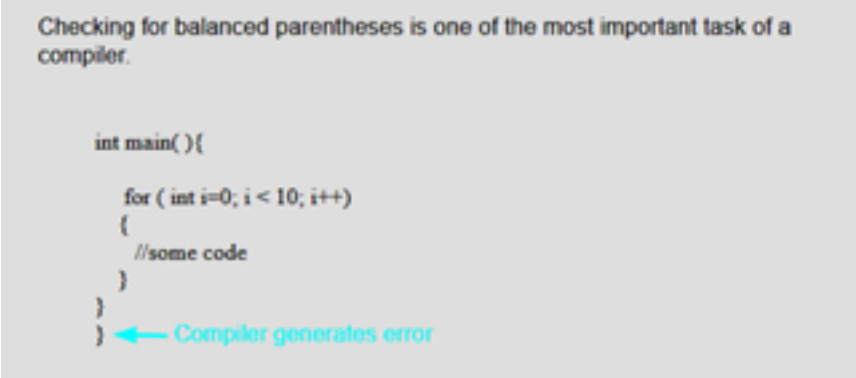
Coding Problem #2: A Deque also known as **double ended queue**, as name suggests is a special kind of **queue** in which insertions and deletions can be done at the last as well as at the beginning. The deque has methods such as insertFirst(), insertLast(), removeFirst(), and removeLast(), as well as a constructor.

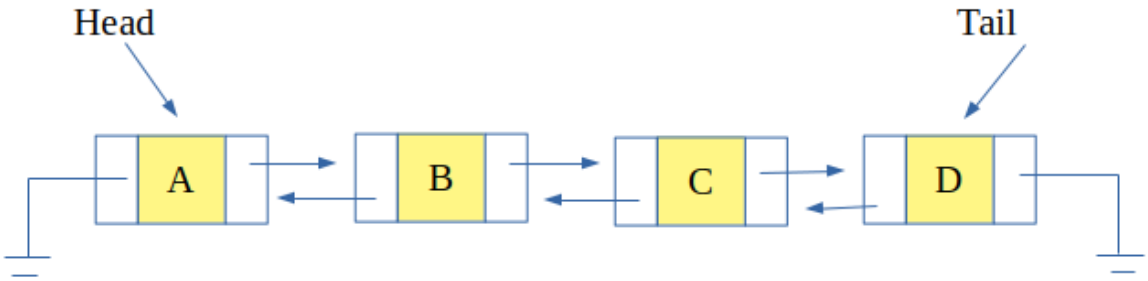
a) Can a deque be efficiently implemented using a singly linked list or a doubly linked list?

b) Can a deque support the stack and queue implementations efficiently? If yes, then write the corresponding stack (push/pop) and queue (enqueue/dequeue) methods using the dequeue data structure. If no, explain why not.

```
public class Stack {

     private Deque;
     …

     void push( String s)
     …

     String pop()
     …

}


public class Queue {

     private Deque;
     …

     void enqueue( String s)
     …

     String dequeue()
     …

}
```

Coding Problem #3: Given an expression string exp , write a program to examine whether the pairs and the orders of "{", "}", "(", ")", "[", "]" are correct in exp. For example, the program should print `true` for exp = "`[()]{}{[()()]()}`" and `false` for exp = "`[(])`"




Head
Tail
A
B
C
D