

## MC 201 Data Structure

### Assignment-1

#### 2024-25 Session

Q.1. Discuss the time complexity of the given codes:

- a.    `for(i=1; i<n; i++)`  
          `i*=k;`
- b.    `int k=0;`  
          `for(i=1; i<=n; i++)`  
              `for(j=1; j<=i; j++)`  
                  `k+=1;`
- c.    `function(int n)`  
          `{`  
              `if(n==1) return;`  
              `for(i=1; i<=n; i++)`  
                  `for(j=1; j<=n; j++)`  
                      `printf("Hello");`  
                  `function(n-3);`  
          `}`
- d.    `function(int n)`  
          `{`  
              `if(n<=1) return;`  
              `for(i=1; i<=3; i++)`  
                  `function(n/3);`  
          `}`

Q.2. Write pseudo-codes for traversing the entire list , searching an element in the list, insertion at the beginning, and deletion at the end of a doubly linked list.

Q.3. Write the pseudo code for counting the number of nodes in any circular linked list.

Q.4. Write the pseudo code to determine whether the single linked list contains a cycle. If the cycle exists in the single linked list, then your algorithm should also give as an output the length of that cycle.

Q.5. Explain how the given infix expression:  $(A+B)*(C-D)*((E+F)*G)$  is converted to postfix expression with the help of stacks. Also determine how that postfix expression is evaluated with stacks.

Q.6. Write the pseudo code that reverses the elements of a stack. Use only the push and pop operations of the stack to reverse the elements of a stack. No additional data structure should be used other than the given stack.

Q.7. Write the pseudo codes to implement one queue using two stacks.

Q.8. Write the pseudo codes to implement one stack using two queues.

Q.9. What does `printf("%-10s", "ABDUL");` display? Explain.

Q.10. What set of conditions is necessary and sufficient for a sequence of insert and remove operations on a single empty queue to leave the queue empty without causing underflow? What set of conditions is necessary and sufficient for such a sequence to leave a nonempty queue unchanged?

Q.11. If an array holding a queue is not considered circular, then each remove operation must shift down every remaining element of the queue. An alternative method is to postpone shifting until rear equals the last index of the array. When that situation occurs and an attempt is made to insert an element into the queue, the entire queue is shifted down, so that the first element of the queue is in position 0 of the array. What are the advantages of this method over performing a shift at each remove operation? What are the disadvantages?