QUEUE

1. A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as a ?

(a) Queue
(b) Stack
(c) Tree
(d) Linked list

2. The data structure required for Breadth First Traversal on a graph is?

(a) Stack
(b) Array
(c) Queue
(d) Tree

3. Let the following circular queue can accommodate maximum six elements with the following data:

front = 2 rear = 4
queue = _____; L, M, N, ___, ___

What will happen after ADD O operation takes place?

a) front = 2 rear = 5
queue = _____; L, M, N, O, ___

b) front = 3 rear = 5
queue = L, M, N, O, ___

c) front = 3 rear = 4
queue = _____; L, M, N, O, ___

d) front = 2 rear = 4
queue = L, M, N, O, ___

4. A queue is a ?

(a) FIFO (First In First Out) list
(b) LIFO (Last In First Out) list
(c) Ordered array
(d) Linear tree

5. If the elements “A”, “B”, “C” and “D” are placed in a queue and are deleted one at a time, in what order will they be removed?

(a) ABCD
(B) DCBA
(c) DCAB
(d) ABCD

6. In linked list implementation of a queue, where does a new element be inserted?

(a) At the head of link list
(b) At the tail of the link list
(c) At the centre position in the link list
(d) None

7. Suppose implementation supports an instruction REVERSE, which reverses the order of elements on the stack, in addition to the PUSH and POP instructions. Which one of the following statements is TRUE with respect to this modified stack?

(a) A queue cannot be implemented using this stack.

(b) A queue can be implemented where ENQUEUE takes a single instruction and DEQUEUE takes a sequence of two instructions.

(c) A queue can be implemented where ENQUEUE takes a sequence of three instructions and DEQUEUE takes a single instruction.

(d) A queue can be implemented where both ENQUEUE and DEQUEUE take a single instruction each.

8. Following is C like pseudo code of a function that takes a Queue as an argument, and uses a

stack S to do processing.

```
void fun(Queue *Q)
{
   Stack S;  // Say it creates an empty stack S
    // Run while Q is not empty
   while (!isEmpty(Q))
{
      // deQueue an item from Q and push the dequeued item to S
      push(&S, deQueue(Q));
}
     // Run while Stack S is not empty
     while (!isEmpty(&S))
```

```
  {
    // Pop an item from S and enqueue the poppped item to Q
    enQueue(Q, pop(&S));
   }
}
```

What does the above function do in general?

(a) Removes the last from Q
(b) Keeps the Q same as it was before the call
(c) Makes Q empty
(d) Reverses the Q

9. Which one of the following is an application of Queue Data Structure?

(a) When a resource is shared among multiple consumers.
(b) When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes
(c) Load Balancing
(d) All of the above

10. How many stacks are needed to implement a queue. Consider the situation where no other data structure like arrays, linked list is available to you.

(a) 1 (b) 2
(c) 3 (d) 4

11. How many queues are needed to implement a stack. Consider the situation where no other data structure like arrays, linked list is available to you.

(a) 1 (b) 2
(c) 3 (d) 4

12. Which of the following is true about linked list implementation of queue?

(a) In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.

(b) In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from the beginning.

(c) Both of the above

(d) None of the above

13. Suppose a circular queue of capacity (n – 1) elements is implemented with an array of n elements. Assume that the insertion and deletion operation are carried out using REAR and FRONT as array index variables, respectively. Initially, REAR = FRONT = 0. The conditions to detect queue full and queue empty are

(a) Full: (REAR+1) mod n == FRONT, empty: REAR == FRONT
(b) Full: (REAR+1) mod n == FRONT, empty: (FRONT+1) mod n == REAR
(c) Full: REAR == FRONT, empty: (REAR+1) mod n == FRONT
(d) Full: (FRONT+1) mod n == REAR, empty: REAR == FRONT

14. An implementation of a queue Q, using two stacks S1 and S2, is given below:

```
void insert(Q, x) {
push (S1, x);
}
void delete(Q){
  if(stack-empty(S2)) then
    if(stack-empty(S1)) then {
       print("Q is empty");
       return;
    }
    else while (!(stack-empty(S1))){
       x=pop(S1);
       push(S2,x);
    }
  x=pop(S2);
}
```

Let n insert and m (<=n) delete operations be performed in an arbitrary order on an empty queue Q . Let x and y be the number of push and pop operations performed respectively in the process. Which one of the following is true for all m and n?

(a) n+m <= x < 2n and 2m <= y <= n+m
(b) n+m <= x < 2n and 2m<= y <= 2n
(c) 2m <= x < 2n and 2m <= y <= n+m
(d) 2m <= x <2n and 2m <= y <= 2n

15. Consider the following operation along with Enqueue and Dequeue operations on queues, where k is a global parameter.

```
MultiDequeue(Q){
  m = k
  while (Q is not empty and m  > 0) {
    Dequeue(Q)
    m = m - 1
  }
}
```

What is the worst case time complexity of a sequence of n MultiDequeue() operations on an initially empty queue? (GATE CS 2013)

(a) theta(n)
(b) theta(n+k)
(c) theta(n^2)
(d) theta(nk)

16. Consider the following pseudo code. Assume that IntQueue is an integer queue. What does the function fun do?

```
void fun(int n)
{
   IntQueue q = new IntQueue();
   q.enqueue(0);
   q.enqueue(1);
   for (int i = 0; i < n; i++)
   {
      int a = q.dequeue();
      int b = q.dequeue();
      q.enqueue(b);
```

```
        q.enqueue(a + b);
        print(a);
       }
}
```

(a) Prints numbers from 0 to n-1
(b) Prints numbers from n-1 to 0
(c) Prints first n Fibonacci numbers
(d) Prints first n Fibonacci numbers in reverse order

17. A circular queue is implemented using an array of size 10. The array index starts with 0, front is 6, and rear is 9. The insertion of next element takes place at the array index.

(a) 0
(b) 7
(c) 9
(d) 10