



- 1- Consider the following statements:

```
Array Queue<int> queue = new Array Queue();
int x, y;
```

Show what is output by the following segment of code:

```
x = 4;
y = 5;
queue.enqueue(x);
queue.enqueue(y);
x = queue.front();
queue.dequeue();
queue.enqueue(x + 5);
queue.enqueue(16);
queue.enqueue(x);
queue.enqueue(y - 3);
system.out.println( "Queue Elements: ");
while (!queue.isEmptyQueue())
{
    system.out.println(queue.front());
    queue.dequeue();
}
```

- 2- What is the output of the following program segment?

```
linkedQueue<int> queue = new linkedQueue();
queue.enqueue(10);
queue.enqueue(20);
cout << queue.front() << endl;
queue.dequeue();
queue.enqueue(2 * queue.back());
queue.enqueue(queue.front());
queue.enqueue(5);
queue.enqueue(queue.back() - 2);
linkedQueue<int> tempQueue = new linkedQueue();
tempQueue = queue;
while (!tempQueue.isEmptyQueue())
{
    system.out.println( tempQueue.front());
    tempQueue.dequeue();
}
system.out.println( queue.front());
system.out.println(queue.back());
```

- 3- Consider the following statements:

```
ArrayStack<int> stack = new ArrayStack();
ArrayQueue<int> queue = new ArrayQueue();
int x;
```

Suppose the input is:



14 8 14 22 64 35 19 32 7 11 13 30 -999

Show what is written by the following segment of code:

```
stack.push(0);
queue.enqueue(0);
system.out.println( x);
while (x != -999)
{
    switch (x % 4)
    {
        case 0:
            stack.push(x);
            break;
        case 1:
            if (!stack.isEmptyStack())
            {
                system.out.println( "Stack Element = " );
                system.out.println( stack.top());
                stack.pop();
            }
        else
            system.out.println( "Sorry, the stack is empty." );
            break;
        case 2:
            queue.enqueue(x);
            break;
        case 3:
            if (!queue.isEmptyQueue())
            {
                system.out.println( "Queue Element = " );
                system.out.println( queue.front());
                queue.dequeue();
            }
            else
                system.out.println( "Sorry, the queue is empty." );
                break;
    } //end switch
    system.out.println( x);
} //end while
system.out.println( "Stack Elements: ");
while (!stack.isEmptyStack())
{
    system.out.println( stack.top() );
    stack.pop();
}
system.out.println( "Queue Elements: ");
while (!queue.isEmptyQueue())
{
```



```

system.out.println( queue.front() );
queue.dequeue();
}

```

- 4- Suppose that queue is a queueType object and the size of the array implementing queue is 100. Also, suppose that the value of queueFront is 50 and the value of queueRear is 99.
  - a- What are the values of queueFront and queueRear after adding an element to queue?
  - b- What are the values of queueFront and queueRear after removing an element from queue?
- 5- Suppose that queue is a queueType object and the size of the array implementing queue is 100. Also, suppose that the value of queueFront is 99 and the value of queueRear is 25.
  - a- What are the values of queueFront and queueRear after adding an element to queue?
  - b- What are the values of queueFront and queueRear after removing an element from queue?
- 6- Suppose that queue is a queueType object and the size of the array implementing queue is 100. Also, suppose that the value of queueFront is 25 and the value of queueRear is 75.
  - a- What are the values of queueFront and queueRear after adding an element to queue?
  - b- What are the values of queueFront and queueRear after removing an element from queue?
- 7- Suppose that queue is a queueType object and the size of the array implementing queue is 100. Also, suppose that the value of queueFront is 99 and the value of queueRear is 99.
  - a- What are the values of queueFront and queueRear after adding an element to queue?
  - b- What are the values of queueFront and queueRear after removing an element from queue?
- 8- Write a function, **reverseQueue**, that takes as a parameter a queue object and uses a stack object to reverse the elements of the queue.
- 9- Suppose an initially empty queue  $Q$  has performed a total of 32 enqueue operations, 10 first operations, and 15 dequeue operations, 5 of which returned null to indicate an empty queue. What is the current size of  $Q$ ?
- 10- What values are returned during the following sequence of deque (*double ended queue*) ADT operations, on an initially empty deque? addFirst(3), addLast(8), addLast(9), addFirst(1), last( ), isEmpty( ), addFirst(2), removeLast( ), addLast(7), first( ), last( ), addLast(4), size( ), removeFirst( ), removeFirst( ).

**Good Luck**