

1. Suppose `queue<int> q` contains 6 elements 1, 2, 3, 4, 5, 6 (enqueued in that order). What is the result of executing the following code snippet? (Assume member function `front()` returns the value found at the front of the queue without removing it.)

```
for(int i = 1; i < 7; i++)
{
    if(i > 0) {
        q.enqueue(q.front());
        q.dequeue();
    }
}
```

- A. the front half of `q` contains even elements and the back half of `q` contains odd elements.
- B. even numbers in `q` are reversed.
- C. odd numbers in `q` are reversed.
- D. elements in the front half of the original `q` are now in the back half.
- E. **[Correct Answer]** **[Your Answer]** `q` remains the same.

2. What is the result of executing the following code snippet?

Assume all required libraries are included and no compile-time/runtime errors occur.

```
int main()
{
    list<int>
    myList;
    for (int
    i=1; i<6; i++)
    myList.push_back(i);

    for (list<int>::iterator it = myList.begin(); it != myList.end(); it++ )
        *it = *it - 2;

    for (list<int>::iterator it = myList.begin(); it !=
    myList.end(); it++ )
        cout << *it << " ";

    return
    0; }
```

- A. **[Your Answer]** -1 0 1 2
- B. 1 2 3 4
- C. None of the other options is correct.
- D. 1 2 3 4 5
- E. **[Correct Answer]** -1 0 1 2 3

3. In implementing Queue ADT, using which of the following data structure gives best asymptotic runtime for enqueue and dequeue? (Assume best possible implementation for queue using provided data structure)

- A. **[Your Answer]** Singly linked list with head and tail pointer.
- B. Doubly linked list with head pointer only.
- C. Doubly linked list with head and tail pointer.
- D. Singly linked list with head pointer only.
- E. **[Correct Answer]** Exactly two of the other options are correct.

4. We have implemented the Stack ADT as an array. Every time the array is full, you resize the array creating a new array that can hold ten more elements than the previous array and copy values over from the old array. What is the total running time for  $n$  pushes to the stack. A.  $O(1)$ .

- B. **[Correct Answer]**  $O(n^2)$ .
- C.  $O(\log n)$ .
- D.  $O(n \log n)$ .
- E. **[Your Answer]**  $O(n)$ .

5. Suppose we have implemented the Queue ADT as a singly-linked-list with head and tail pointers and no sentinels. Which of the following best describe the tightest running times for the functions enqueue and dequeue, assuming there are  $O(n)$  items in the list, and that the rear of the queue is at the head of the list?

- A.  $O(n)$  for both.
- B. **[Correct Answer]**  $O(1)$  for enqueue and  $O(n)$  for dequeue.
- C. **[Your Answer]**  $O(1)$  for both.
- D. None of the options is correct
- E.  $O(n)$  for enqueue and  $O(1)$  for dequeue.