

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

2. 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 front of the queue is at the head of the list?

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

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

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 four times as many elements as the previous array and copy values over from the old array. What is the total running time for n pushes to the stack.

- A. Correct Answer Your Answer $O(n)$.
- B. $O(1)$.
- C. $O(n \log n)$.
- D. $O(\log n)$.
- E. $O(n^2)$.

5. In implementing Stack ADT, using which of the following data structure gives best asymptotic runtime for push and pop? (Assume best possible implementation for stack using provided data structure)

- A. Singly linked list with head and tail pointer.
- B. Singly linked list with head pointer only.
- C. None of the options provide the best asymptotic runtime.
- D. Array (size of array larger than possible elements in stack).
- E. Correct Answer Your Answer All options provide the same runtime.