NetID: roatis2 QuizID: 68447 Score: 4 / 5 Answer Source: PrairieLearn

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

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

3. Suppose we have implemented the Queue ADT as a singly-linked-list with only `head` pointer and no sentinels. Which of the following best describe the 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. $O(1)$ for both.
B. None of the options is correct
C. [Correct Answer] [Your Answer] $O(n)$ for `enqueue` and $O(1)$ for `dequeue`.
D. $O(1)$ for `enqueue` and $O(n)$ for `dequeue`.
E. $O(n)$ for both.

4. Consider this (and assume all STL classes are available):

```
struct animal {
    string name;
    string food;
    animal(string n="blob", string f="you"): name(n), food(f) { }
};

int main() {
    animal g("giraffe","leaves"), b("bear");
    list<animal *> zoo;
    zoo.push_back(&g); zoo.push_back(&b); //STL list insertAtEnd
    for(list<animal *>::iterator it = zoo.begin(); it != zoo.end(); it++)
        cout << (*it)->name << " " << (*it)->food << endl;
}
```

Which of the following describes the result when this function is compiled and run?

A. [Correct Answer] [Your Answer] The code sends "giraffe leaves" and "bear you" to standard out.
B. The code sends "giraffe leaves" to standard out.
C. There is a compiler error because of a type mismatch in the parameterized type for the list class.
D. There is a compiler error because there is no constructor matching the one called for variable b.
E. There is a runtime error because the iterator is dereferenced twice.

5. 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. $O(n \log n)$.
C. $O(1)$.
D. $O(\log n)$.
E. $O(n^2)$.