Mohammed Touhid Chowdhury

HW#3

September 28, 2016

Written part:

1.

Output:

6
2
5
9
7

The line (c.end() – c.begin())/2 gives the iterator to the middle element of the list so c.begin() plus the middle gives an output of the 7th element which is 6. The next line that’s printed is itr1 which is the beginning element plus 2 spots ahead which is 2. The line c.erase(itr2); erases the current element at itr2 and shifts the remaining elements one position above. When itr2 is printed it goes the the element after it was previously holding because the old one was deleted so 5 is printed. Itr3 is also moved ahead because of the itr2 being deleted so it prints 9. This time when the c.begin() + middle is printed, it prints the element ahead which is 7.

2. The worst case running time is **O(n).**

3.

Vector<int>::iterator itrStart = **a.begin();**
Vector<int>::iterator itrMid = **a.begin() + ((c.end()-c.begin())/2);**
Vector<int>::iterator itrEnd = **a.end();**

4.

(a) **4** is printed

(b) **Doesn’t compile** because the list doesn’t have the random access. Could only use itrb++;

(c) **5** is printed

(d) **Wont compile** because the type of iterator and vector isn’t the same