1. Without using the STL implement a class to maintain a stack of doubles (StackOfDoubles)

a. The class should use the list of doubles you implemented last week, ListOfDoubles internally. (i.e. you don’t have to code a stack from scratch).

It will have the usual stack methods:

- push() //pushes a double onto the stack

- pop() // removes the top item from the stack without returning it

- top() //returns the top item from the stack without removing it

b. The stack should also facilitate an overloaded insertion operator to insert the contents of the stack (tab separated) into an output stream. The signature of this overloaded operator is:

ostream & operator<<(ostream& str, const StackOfDoubles &stackobj)

c. Write a program which fully tests your class.

2. Using the STL implement a queue of doubles by

– Composition, using an STL list (list<double>) to hold the data

<http://www.cplusplus.com/reference/queue/queue/>

Now if we wanted to use an STL vector (vector<double>) to hold the data, we cannot as it does not have a pop\_front() operation; such an operation would be slow, as it would have to move all the remaining elements. However if we want to make a priority queue we can use a vector. Therefore now.

3. Using the STL implement a priority queue of doubles by

– Composition, using an STL vector (vector<double>) to hold the data.

<http://www.cplusplus.com/reference/queue/priority_queue/>

<http://www.cplusplus.com/reference/queue/priority_queue/priority_queue/>