Questions from asg 2

1. One of the ways people deal with the performance issues of linear searching a list is to build a self-organizing list.    This data structure works just like the list data structure for it's public interface except that it defines a  find member function that searches the list and, if it finds something, moves that entry to the front of the list.

Define a C++ template class that is a wrapper around the C++ STL list template class that implements this data type.   Note that this is going to be an exercise in the use of iterators and you having to do a deep dive into the methods in the interface of the C++ STL list template class.

2. We introduced you to the concept of smart pointers in lecture.    Use the C++ STL smart pointer classes to implement a Tree template class, as we discussed in the lecture slides.   Provide a test program that constructs a BST with 1000 random numbers and then searchs for that 100 numbers in that tree.

3. One of the common ways we had before computers for organizing a term paper was to write out our notes on index cards.  Each card would be labeled with a keyword and an unique index number.  We would arrange and sort the cards to organize our thoughts before starting to write the paper.

Design and construct a program that will allow you enter and search a deck of index cards.   Use the C++ STL as much as possible in your design and implementation.

Document your design using UML class, object, and sequence diagrams.

Be certain to generate a set of unit tests that provide adequate test coverage for your program.