**C++ Workshop – 150018**

**Homework Assignment #5**

**Linked Lists**

1. Complete the List class definition discussed in lecture to handle lists whose elements are sorted in strictly decreasing order (every element is less than its predecessor). You may find it helpful to use the class defined in the course handbook. Add the following methods:
   1. operator=: deep assignment of elements of a list to another list retaining same order of elements
   2. operator<<: prints all elements of the sorted list.
   3. operator>>: inputs sorted elements. Note, the input is terminated when it receives a value that is not strictly less than its predecessor.
   4. insert(int key): inserts the given integer into its appropriate place in the sorted list. At this point, the list is in non-ascending order (this allows duplicate entries).
   5. remove(int key): removes the element whose value matches the given key. The method preserves sorted order. If more than one element matches the key, then only the first one is removed. If the key value is not found, the function throws the exception "value not found".

Use the following main program to check your functions:

int main()

{

List lst;

int choice,val;

cout<<"enter the list values\n";

cin>>lst;

cout<<"choose 0-2\n";

cin>>choice;

while (choice)

{

switch (choice)

{

case 1:cout<<"enter a value to insert\n";

cin>>val;

lst.insert(val);

break;

case 2:cout<<"enter a value to remove\n";

cin>>val;

try{

lst.remove(val);

}

catch(char \* msg)

{

cout<<msg<<endl;

}

break;

default:cout<<"ERROR\n";

}

cout<<lst<<endl;

cout<<"choose 0-2\n";

cin>>choice;

}

return 0;

}

Example:

enter the list values

3 2 1 1

choose 0 - 2

2

enter a value to remove

2

3 1

choose 0 - 2

1

enter a value to insert

4

4 3 1

choose 0 - 2

1

enter a value to insert

2

4 3 2 1

choose 0 - 2

0

1. Using the sorted List class you defined above, write the following three **global** functions. (Note, you should **not** define these functions as friends of the class, but rather have them use public helper methods defined in the class.)
   1. **merge** that merges two lists. The function receives two sorted lists, lst1 and lst2, each of type List, and returns a new list that combines the two lists in non-asscending order. Note: the newly created list may have duplicate entries.  
        
      
   2. **makeSet** that converts a list into a set with no duplicate entries. The function receives a list of integers in non-asscending order and converts it into an ordered set having no duplicate values. The set should be in strictly ascending order.

7

5

4

3

1

* 1. Use the following main program to check your functions:

#include <iostream>

#include "List.h"

using namespace std;

//הגדרת ומימוש הפונקציות

int main()

{

List lst1, lst2, mergedList;

cout<<"enter sorted values for the first list:"<< endl;

cin>>lst1;

cout<<"enter sorted values for the second list:"<< endl;

cin>>lst2;

mergedList = merge(lst1,lst2);

cout <<"the new merged list: " << mergedList <<endl;

makeSet(mergedList);

cout<<"the new merged set: " << mergedList << endl;

reverse(mergedList);  
cout<<"the new merged reverse: " << mergedList << endl;

return 0;

}

enter sorted values for the first list :

6 5 4 3 2 1 9

enter sorted values for the second list :

7 5 4 3 9

the new merged list : 7 6 5 5 4 4 3 3 2 1

the new merged set : 7 6 5 4 3 2 1

the new merged reverse : 1 2 3 4 5 6 7

Press any key to continue . . .

Example: