North South University

Department of Electrical and Computer Engineering

CSE 225L.13 (Data Structures and Algorithms Lab)

Lab 5: Unsorted Lists (Array Based)

Instructor: Syed Shahir Ahmed Rakin, Arfana Rahman

Objective:

- Learn how the Unsorted Lists work when made with arrays.

Remember the Arrays:

An array is a data structure representing a collection of the same data types. The process of declaring arrays is given below:

data_type info[size]

Unsorted List:

An unsorted list is an abstract data structure where the values are given in an unsorted manner, here, we are using an array to make an unsorted list. An example of an unsorted list is given as follows:

5	3	7	6	1

Here, you can see that the unsorted list is represented by an array, and you can delete and retrieve any item as you can see fit.

Prototype of Unsorted List:

The header and source file of the Array-based Unsorted List is given as follows.

unsortedtype.h

```
#ifndef UNSORTEDTYPE_H_INCLUDED
#define UNSORTEDTYPE_H_INCLUDED
const int MAX_ITEMS = 5;
template <class ItemType>
class UnsortedType{
public :
    UnsortedType(); void MakeEmpty();
    bool IsFull(); int LengthIs();
    void InsertItem(ItemType);
    void DeleteItem(ItemType);
    void RetrieveItem(ItemType&, bool&);
    void ResetList();
    void GetNextItem(ItemType&);
private:
    int length; ItemType info[MAX_ITEMS];
    int currentPos;
};
#endif // UNSORTEDTYPE_H_INCLUDED
```

```
template <class ItemType>
voidUnsortedType<ItemType>::RetrieveItem(
ItemType&
item, bool &found)
{
    int location = 0;
    bool   moreToSearch   =   (location   <
length);
    found = false;
    while (moreToSearch && !found)
    {
        if(item == info[location])
        {
            found = true;
            item = info[location];
        } else {
            location++;
            moreToSearch   =   (location   <
length);
        }
    }
}
```

unsortedtype.cpp

```
#include "UnsortedType.h"
template <class ItemType>
UnsortedType<ItemType>::UnsortedType()
{
    length = 0;
    currentPos = -1;
}

template <class ItemType>
void   UnsortedType<ItemType>::MakeEmpty()
{
    length = 0;
}

template <class ItemType>
bool UnsortedType<ItemType>::IsFull()
{
    return (length == MAX_ITEMS);
}

template <class ItemType>
int UnsortedType<ItemType>::LengthIs()
{
    return length;
}

template <class ItemType>
void UnsortedType<ItemType>::ResetList()
{
    currentPos = -1;
}

template <class ItemType>
void
UnsortedType<ItemType>::GetNextItem(ItemTy
pe&
item)
{
    currentPos++;
    item = info [currentPos] ;
}

template <class ItemType>
void
UnsortedType<ItemType>::InsertItem(ItemTy
pe
item){
    info[length] = item;
    length++;
}

template <class ItemType>
void
UnsortedType<ItemType>::DeleteItem(ItemTy
pe
item)
{
    int location = 0;
    while (item != info[location]){
        location++;
    }

    info[location] = info[length - 1];
    length--;
}
```

Tasks:

Generate the **driver file (main.cpp)** where you perform the following tasks. However, there is a restriction that *<u>you cannot make any changes to the header file or the source file</u>*.

Operation to Be Tested and Description of Action	**Input Values**	**Expected Output**
▪ Create a list of integers		
▪ Insert four items	5 7 6 9	
▪ Print the list		5 7 6 9
▪ Print the length of the list		4
▪ Insert one item	1	
▪ Print the list		5 7 6 9 1
▪ Retrieve 4 and print whether found or not		Item is not found
▪ Retrieve 5 and print whether found or not		Item is found
▪ Print if the list is full or not		List is full
▪ Delete 5		
▪ Print if the list is full or not		List is not full
▪ Delete 1		

▪ Print the list		7 6 9
▪ Write a class studentInfo that represents a student record. It must have variables to store the student ID, student's name, and student's CGPA. It also must have a function to print all the values. You will also need to overload a few operators.		
▪ Create a list of objects of class studentInfo.		
▪ Insert 5 student records	15234 Yuji Itadori 2.6 13732 Megumi Fushigoro 3.9 13569 Nobura Kugisaki 1.2 15467 Satoru Gojo 4.0 16285 Ryomen Sukuna 3.1	
▪ Print the list		15234, Yuji Itadori, 2.6 13732, Megumi Fushigoro, 3.9 13569, Nobura Kugisaki, 1.2 15467, Satoru Gojo, 4.0 16285, Ryomen Sukuna, 3.1