Data Structures Lab

Searching and Sorting of Arrays

Objective: Complete a class that implements an Unordered array (Outlined below).

|  |
| --- |
| UnOrderedArray |
| m\_array (type int[])  maxSize (type int)  numElements (type int) |
| Constructor (int size)  addLast (int item)  removeLast ()  efficientRemove(int index)  remove/ByIndex( int index)  linearSeach(int item)  listItems ()  insertionSortAsc()  selectionSortAsc() |

Main functions defined below:

**addLast** : takes the integer to be added as a parameter and inserts it at the end of the array.Duplicates are allowed.

**removeLast** : removes the last item from the array

**removeByIndex**: takes the index of the item to be removed as the parameter and removes it by shifting items down from the right.

**efficientRemove** : takes the last item and swaps it with the item in the index to be removed.

**listItems** : prints out all items in the array

**linearSearch** : takes an item as a parameter and returns the index (position) in the array it was found, if not found return -1

**insertionSortAsc:** sorts the array using insertion sort in ascending order

**selectionSortAsc :** sorts the array using selection sort in descending order

Part 2

Objective: Complete a class that implements an Ordered array (Outlined below).

|  |
| --- |
| OrderedArray |
| m\_array (type int[])  maxSize (type int)  numElements (type int) |
| Constructor (int size)  addInOrder (int item)  removeByIndex( int index)  removeItem (int item)  linearSeach(int item)  listItems ()  binarySearch(int item) |

**addInOrder** : takes the integer to be added as a parameter and inserts it **in order** in the array. No duplicates are allowed.

**removeByIndex**: takes the index of the item to be removed as the parameter and removes it by shifting items down from the right.

**removeItem**: takes an item to be removed as the parameter and removes it by shifting items down from the right.

**listItems** : prints out all items in the array

**linearSearch** : takes an item as a parameter and returns the index (position) in the array it was found, if not found return -1

**binarySearch** : takes an item as a parameter and returns the index (position) in the array it was found, if not found return -1