**North South University - Spring 2023**

Course: CSE225L Assessment: Lab 4

Section: 6 NSU ID: 2211424642 Name: Joy Kumar Ghosh

// Task-1 Codes

|  |
| --- |
| #ifndef UNSORTEDTYPE\_H  #define UNSORTEDTYPE\_H  #include <iostream>  using namespace std;  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 |
| #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(ItemType &item){  currentPos++;  item = info[currentPos];  }  template <class ItemType>  void UnsortedType<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);  }  }  }  template <class ItemType>  void UnsortedType<ItemType>::InsertItem(ItemType item){  info[length] = item;  length++;  }  template <class ItemType>  void UnsortedType<ItemType>::DeleteItem(ItemType item){  int location = 0;  while (item != info[location])  location++;  info[location] = info[length - 1];  length--;  } |
| #include "unsortedtype.cpp"  int main()  {  //creating a integer list  UnsortedType<int> arrayList;  //supporting variable  int item, i;  bool isFound;  //taking 4 input and inserting in list  for(int i = 0; i < 4; i++){  cin >> item;  arrayList.InsertItem(item);  }  //printing list  for(i = 0, arrayList.ResetList(); i < arrayList.LengthIs(); i++){  arrayList.GetNextItem(item);  cout << item << " ";  }  cout << endl;  //printing list length  cout << arrayList.LengthIs() << endl;  //inserting one item by user input  cin >> item;  arrayList.InsertItem(item);  //printing list  for(i = 0, arrayList.ResetList(); i < arrayList.LengthIs(); i++){  arrayList.GetNextItem(item);  cout << item << " ";  }  cout << endl;  //retrieving 4  item = 4;  arrayList.RetrieveItem(item, isFound);  if(isFound)  cout << "Item is found" << endl;  else  cout << "Item is not found" << endl;  //retrieving 5  item = 5;  arrayList.RetrieveItem(item, isFound);  if(isFound)  cout << "Item is found" << endl;  else  cout << "Item is not found" << endl;  //retrieving 9  item = 9;  arrayList.RetrieveItem(item, isFound);  if(isFound)  cout << "Item is found" << endl;  else  cout << "Item is not found" << endl;  //retrieving 10  item = 10;  arrayList.RetrieveItem(item, isFound);  if(isFound)  cout << "Item is found" << endl;  else  cout << "Item is not found" << endl;  //checking list full or not  if(arrayList.IsFull())  cout << "List is full" << endl;  else  cout << "List is not full" << endl;  //deleting item 5  arrayList.DeleteItem(5);  //checking length full or not  if(arrayList.IsFull())  cout << "List is full" << endl;  else  cout << "List is not full" << endl;  //deleting item 1  arrayList.DeleteItem(1);  //printing list  for(i = 0, arrayList.ResetList(); i < arrayList.LengthIs(); i++){  arrayList.GetNextItem(item);  cout << item << " ";  }  cout << endl;  //deleting item 6  arrayList.DeleteItem(6);  //printing list  for(i = 0, arrayList.ResetList(); i < arrayList.LengthIs(); i++){  arrayList.GetNextItem(item);  cout << item << " ";  }  cout << endl;  return 0;  } |
|  |

// Task-2 Codes

|  |
| --- |
|  |
|  |
|  |

// Task-3 Codes

|  |
| --- |
|  |
|  |
|  |

// Task-4 Codes

|  |
| --- |
|  |
|  |
|  |

// Task-5 Codes

|  |
| --- |
|  |
|  |
|  |