//Name : Yashvi Shah

//SID : 201701045

//Assignment No : 4

//Description : Implement an arraylist using following classes

//Last changed : 19/02/2018

#include<iostream>

#include<stdlib.h>

using namespace std;

template <typename E>

class List

{

void operator = (const List&) {}

List(const List&) {}

public:

//constructors and destructors

List() {}

virtual ~List() {}

//Return the number of elements in the List

virtual int NumberOfElements() const = 0;

//Return the Location of current element

virtual int CurrentLocation() const = 0;

//Return the data of current element

virtual const E& getData() const = 0;

//clear all the data from the List

virtual void clear() = 0;

//insert a data in the List at current location

virtual void insert(const E& data) = 0;

//insert a data in the end of the List

virtual void append(const E& data) = 0;

//delete a data at the current Location

virtual E remove() = 0;

//set the current Location to the start of the List

virtual void setListLocationToStart() = 0;

//set the current Location to the end of the List

virtual void setListLocationToEnd() = 0;

//set the current Location to the element on left side

// of current element

virtual void previousElement() = 0;

//set the current Location to the element on left side

// of current element

virtual void nextElement() = 0;

//Set current Location to a new Location

virtual void setToNewLocation(int location) = 0;

//display the elements

virtual void display() = 0;

};

case 6:cout<<"ENTER THE ELEMENT TO BE APPENDED AT THE LAST :"<<endl;

cin>>x;

A.append(x);

A.display();

break;

case 7:cout<<"THE REMOVED ELEMENT IS"<<A.remove()<<endl;

A.display();

break;

case 8:A.setListLocationToStart();

break;

case 9:A.setListLocationToEnd();

break;

case 10:A.previousElement();

break;

case 11:A.nextElement();

break;

case 12:cout<<"ENTER THE LOCATION YOU WANT TO SET AS CURRENT LOCATION : "<<endl;

cin>>x;

A.setToNewLocation(x);

break;

case 13:exit(1);

break;

default:cout<<"INVALID INPUT"<<endl;

break;

}

}

return 0;

}