Lab 02

#include <iostream>

using namespace std;

/\* write a Function name Linear Fuction that will an arry,

its size and element to search as parametrs, this function should display found

if element exist otherwise not found. \*/

void initArry(int arr[], int size, int element);// intilize

void linearSearch(int arr[], int size, int element);// Searching

void binarysearch(int arr[], int size, int element);// binary Search

void disArry(int arr[], int size);// Display

void eveD(int arr[], int size);// Even

/\* write a fucntion with name swap that will take an arry and to indcies as parameters.

the function should exchange the values of provided indcies.\*/

void swap(int arr[], int indexA, int indexB);// Swap Function

/\* write a with name largest that will take an Arry an size of Arry as a parameters.

The function should return the index of largest element in th Arry.\*/

int largest(int arr[], int size);// Return largest index

int main(int argc, char\*\* argv){

int size=0;

cout<<"Enter size "<<endl;

cin>>size;// Size of arry

int arry[size];

int element;

int A;

int B;

int choice;

do{

cout<<" Press 1 to intilize "<<endl;

cout<<" Press 2 to display an arry "<<endl;

cout<<" Press 3 to linear search in arry "<<endl;

cout<<" Press 4 to Binary search in arry "<<endl;

cout<<" Press 5 to get even numbers "<<endl;

cout<<" Press 6 Swap indices "<<endl;

cout<<" Press 7 to find largest index "<<endl;

cout<<" Press 0 to exit from code "<<endl;

cin>>choice;

switch(choice)

{

case 1:

cout<<" Providing values of arry "<<endl;

initArry(arry, size, element);// Calling

break;

case 2:

cout<<" Entered arry is "<<endl;

disArry( arry, size);

break;

case 3:

// Element we want to search

cout<<"Enter element to search "<<endl;

cin>>element;

cout<<" Searching of element "<<endl;

linearSearch(arry, size, element); // Calling

break;

case 4:

cout<<"Enter element to search "<<endl;

cin>>element;

binarysearch( arry, size, element);

break;

case 5:

eveD( arry, size);

break;

case 6:

cout<<"Before swap "<<endl;

disArry(arry, size);

cout<<"Enter indices to swap "<<endl;

cin>>A>>B;

swap( arry, A, B);

cout<<"After Swap "<<endl;

disArry(arry, size);

break;

case 7:

cout<< largest( arry, size)<<endl;

break;

case 0:

exit(-1);

break;

}// switch

}// do

while(true);

return 0;

}

void initArry(int arr[], int size, int element) // defination of function

{

for (int i=0; i<size; i++)

{

cin>>arr[i];

}

}

void linearSearch(int arr[], int size, int element)

{

int i;

//cout<<" Arry searching "<<endl;

// while (i<size && element!=arr[i]){

// i++;

// }

for ( i=0; i<size && element!=arr[i]; i++)

{

}

if (i>=size)

{

cout<<" Not found "<<endl;

}

else{

cout<<" Found "<<endl;

}

}

void disArry(int arr[], int size){

//cout<<"Element of Arry is \n" ;

for (int i=0; i<size; i++)

{

// cout <<"\n";

cout<<arr[i]<<endl;

}

}

void eveD( int arr[], int size){

for (int i=0; i<size; i++)

{

if (arr[i]%2==0)

{

cout<<"Even no is "<<arr[i]<<endl;

}

else

{

}

}

}

void binarysearch(int arr[], int size, int element){

int beginning =0;

int end =size-1;

int mid = (beginning+end)/2;

while(beginning<=end && element!=arr[mid])

{

// there are still element to explore

// element is not found

if (element< arr[mid])// back

{

end=mid-1;

}

else if(element>arr[mid])// forward

beginning=mid+1;

mid = (beginning+end)/2;// new mid base on nmodified b and e

} //while

if(beginning>end)

{ // check all

cout<<" not Found "<<endl;

}

else

{

cout<<" Found "<<endl;

}

}// binary Search

void swap(int arr[], int indexA, int indexB)

{ int c;

c=arr[indexA];// indexA swap in C

arr[indexA]= arr[indexB];// indexB swap in A

arr[indexB]=c;// c swap in indexB

// cout<<"Swap values is "<<endl;

// cout<<arr[indexA]<<endl;

//cout<<arr[indexB]<<endl;

}// Swap

int largest(int arr[], int size){

// int largestIndex=arr[0];

int largest\_loc=0;

for(int i=1; i<size; i++)

{

if(arr[largest\_loc]<arr[i])

{

largest\_loc=i;

}

}

/\* int index;

for (int i=1; i<size; i++)

{

if(largestIndex<arr[i])

{

largestIndex=arr[i];

index=i;

} \*/

return largest\_loc;

//return index;

}

Lab 03

Link List

Code of class Node

#include<iostream>

using namespace std;

class Node

{ private:

int info; // data Hiding

Node\* next;

public :

Node(int info,Node\* next)

{

this->info=info;

this->next=next;

}

void setInfo(int info);

void setNext(Node \*next);

int getInfo();

Node\* getNext();

void displayNode();

};// end of Node class get

// Its C++ standard to implement attribute of outside of class

void Node:: setInfo(int info) //:: scope resulution operator

{

this->info=info;

}//set info

void Node:: setNext(Node\* next) //:: scope resulution operator

{

this->next=next;

}//set \*next

int Node:: getInfo()

{

return this->info;

}//set

Node\* Node:: getNext()

{

return this->next;

}//set

void Node::displayNode()

{

cout<<"("<<this<<") |"<<info<<"|"<<next<<"|"<<endl;

}

Code of Class SLL

#include<iostream> // Standard headerfile --> <>

#include "Node.h" // LL contain obkj of Node class // is our own headerfiles --> ""

using namespace std;

class SLL

{

private:

Node \*head;// first node

Node \*tail;// last node

public:

SLL()

{

head=0;

tail=0;

}

void setHead(Node\* first);

void setTail(Node \*last);

Node\* getfirst();

Node\* getlast();

void traverse();// vist all one by one's

void addTohead(int element);

}; // end class

void SLL::setHead(Node\* first)

{

head=first;

} // set

void SLL::setTail(Node\* last)

{

tail=last;

} //set

Node\* SLL::getfirst()

{

return head;

}

Node\* SLL::getlast()

{

return tail;

}

void SLL:: traverse()

{

Node\* i=head;

while(i!=0)

{

cout<<i->getInfo()<<endl;

i=i->getNext();

}

}//end traverse

void SLL::addTohead(int element)

{

Node \*n= new Node( element,0);

if(head==0 && tail==0)

{

head=n;

tail=n;

}

else

{

n->setNext(head);

head=n;

}

}

Main Class

#include <iostream>

#include "SLL.h"

using namespace std;

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char\*\* argv) {

SLL list1;

list1.addTohead(17);// /17/0 H,T

list1.addTohead(15);// without new mean compile time call // /15\0xA1/H

list1.addTohead(19);

list1.traverse();

return 0;

}