**UCS540**

**Data Structures and Algorithms**

Submitted by: Ritik Singh (102104089)

Class: 3EE3

**Lab Assignment 2**

**Arrays and Pointers**

**Objectives: Students should be able to perform various operations on array data structures. To explain the intimate relationship between arrays and pointers (along with pointer arithmetics).**

1. **Write a program to check whether a given number is present in an array or not (Linear search).**

**Code:**

#include <iostream>

using namespace std;

int main() {

int n;

int num;

int i;

while(n!=0){

cout<<"enter the size of array:"<<endl;

cin>>n;

if(n==0){

cout<<"program terminated\n"<<endl;

}

int arr[n];

cout<<"enter all numbers:\n";

for(int i=0;i<n;i++){

cin>>arr[i];

}

cout<<"enter the no. to check that is present in array or not:"<<endl;

cin>>num;

bool found=false;

for(int i=0;i<n;i++){

if (arr[i]==num){

found=true;

break;

}

}

if(found){

cout<<num<<"is present in the array\n";}else

{

cout<<num<<"is not present in the array\n";

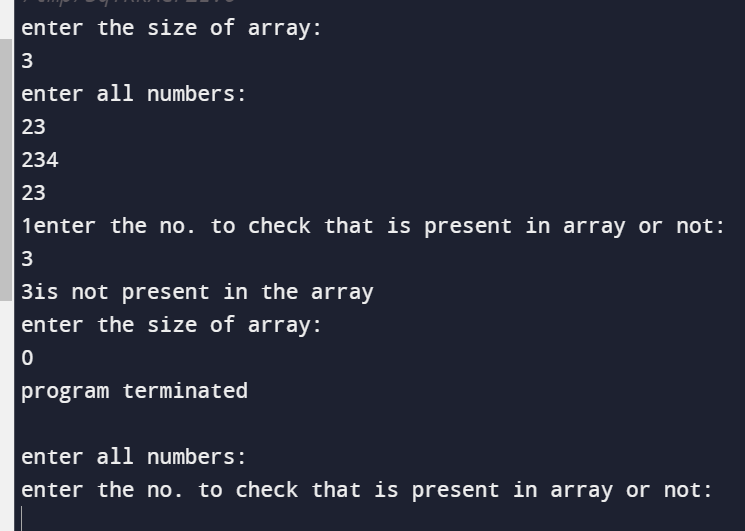
}

}

return 0;

}

**OUTPUT:**

****

1. **Write a program to get second maximum and second minimum elements in an array.**

**Code:**

#include <bits/stdc++.h>

using namespace std;

int main() {

int n;

cout<<"enter size of array;\n"<<endl;

cin>>n;

int arr[n];

cout<<"enter array elements:\n"<<endl;

for(int i=0;i<n;i++){

cin>>arr[i];

}

int firstMax=INT\_MIN, secondMax=INT\_MIN;

int firstMin=INT\_MAX, secondMin=INT\_MAX;

for(int i=0;i<n;i++){

if(arr[i]>firstMax){

secondMax=firstMax;

firstMax=arr[i];

}

else if(arr[i]>secondMax && arr[i]==firstMax){

secondMax=arr[i];

}

if (arr[i]<firstMin){

secondMin=firstMin;

firstMin=arr[i];

}

else if(arr[i]<secondMin && arr[i]!=firstMin){

secondMin=arr[i];

}

}

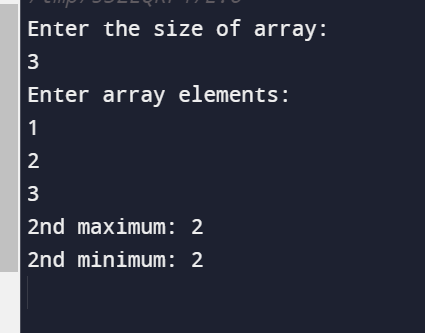
cout<<"2nd maximum"<<secondMax<<"\n";

cout<<"2nd minimum"<<secondMin<<"\n";

return 0;

}

**OUTPUT:**

****

1. **Write a program to perform insertion (any location), deletion (any location) and traversal in an array.**

**a) insert an array**

**Code:**

#include<bits/stdc++.h>

using namespace std;

//question; insert element in array

// say 1,2,3 are elements and 4 is to be inserted at 2 position

//so new array becomes 1,2,4,3

int main(){

int n;

cout<<"enter size of array(n)\n";

cin>>n;

int elements;

int arr[n];

cout<<"enter array elements:"<<endl;

for(int i=0;i<n;i++){

cin>>arr[i];

}

int number,position;

cout<<"enter number to be entered\n";

cin>>number;

cout<<"enter position where no. you want to add\n";

cin>>position;

for(int i=n;i>=position;i--) //array increases its size because number is inserted,and i goes till that position,and value decreases

arr[i+1]=arr[i];//last element shifts further

arr[position]=number;//vacant position occupies the number

cout<<"new array becomes:\n";

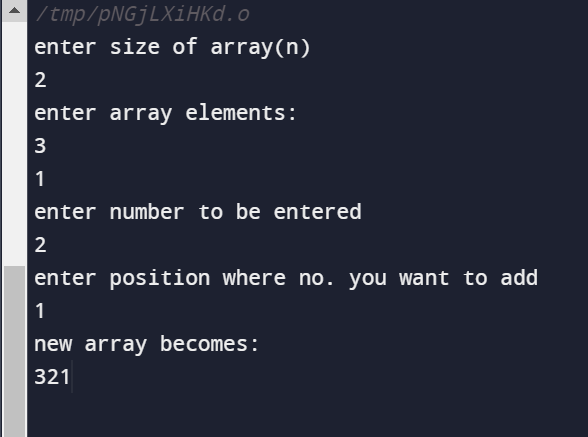
for(int i=0;i<n+1;i=i+1)

cout<<arr[i];

return 0;

}

**OUTPUT:**

****

**B) delete an array:**

#include<bits/stdc++.h>

using namespace std;

//delete an array

// say 1,2,3 numbers, 2 is to be deleted, hence numbers become 1,3

int main(){

int n;

cout<<"enter size of array(n)\n";

cin>>n;

int elements;

int arr[n];

cout<<"enter array elements:"<<endl;

for(int i=0;i<n;i++){

cin>>arr[i];

}

int position;

cout<<"enter position where no. you want to delete\n";

cin>>position;

for(int i=position;i<n;i=i+1) {

arr[i]=arr[i+1];

}

cout<<"new array becomes:\n";

for(int i=0;i<n-1;i=i+1)

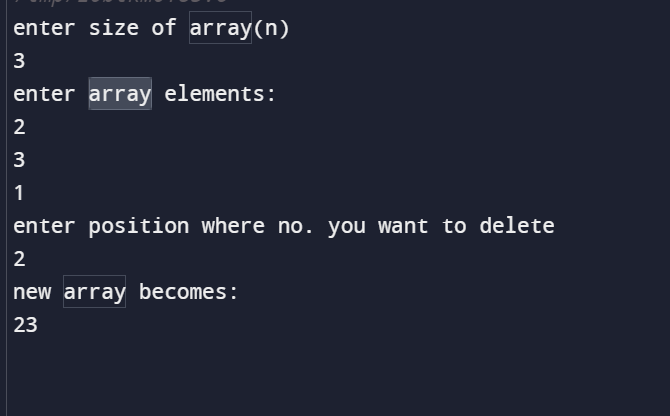
cout<<arr[i];

cout<<"\n";

return 0;

}

**OUTPUT:**

****

**C) Traversal in an array**

#include<iostream>

using namespace std;

int main(){

// tranversal of an array means printing each element of an array seperately

int n;

cout<<"enter size of array(n)\n";

cin>>n;

int elements;

int arr[n];

cout<<"enter array elements:"<<endl;

for(int i=0;i<n;i++){

cin>>arr[i];

}

cout<<"elements of array are:\n";

for(int i=0;i<n;i++){

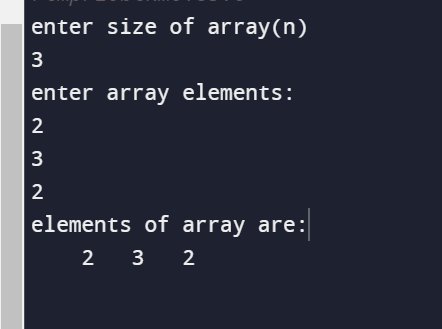
cout<<"\t"<<arr[i];

}

return 0;

}

**OUTPUT:**

****

1. **Write a menu driven program to perform addition, multiplication and subtraction of 2 arrays.**

**Code:**

#include <iostream>

using namespace std;

void performOperation(double array1[], double array2[], int size, int operation) {

double result[size];

switch (operation) {

case 1:

cout << "Addition Result:\n";

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

result[i] = array1[i] + array2[i];

cout << result[i] << " ";

}

break;

case 2:

cout << "Multiplication Result:\n";

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

result[i] = array1[i] \* array2[i];

cout << result[i] << " ";

}

break;

case 3:

cout << "Subtraction Result:\n";

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

result[i] = array1[i] - array2[i];

cout << result[i] << " ";

}

break;

default:

cout << "Invalid operation!\n";

return;

}

cout << endl;

}

int main() {

int size, operation;

cout << "Enter the size of the arrays: ";

cin >> size;

double array1[size], array2[size];

cout << "Enter elements of array 1:\n";

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

cout << "Element " << i + 1 << ": ";

cin >> array1[i];

}

cout << "Enter elements of array 2:\n";

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

cout << "Element " << i + 1 << ": ";

cin >> array2[i];

}

cout << "Menu:\n";

cout << "1. Addition\n";

cout << "2. Multiplication\n";

cout << "3. Subtraction\n";

cout << "Enter the operation number (1/2/3): ";

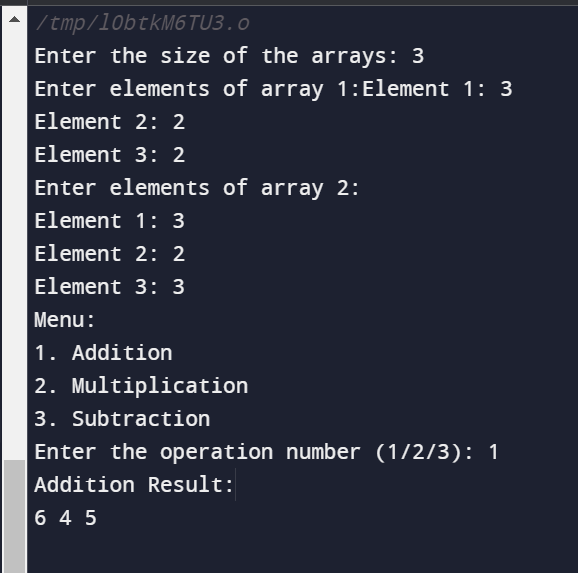
cin >> operation;

performOperation(array1, array2, size, operation);

return 0;

}

**OUTPUT:**

****

1. **Write a program to perform sorting while merging (Merge two sorted arrays into one sorted array).**

**CODE:**

#include <iostream>

#include <vector>

using namespace std;

void merge(vector<int> &arr, int left, int mid, int right)

{

int n1 = mid - left + 1;

int n2 = right - mid;

// Create temporary arrays

vector<int> L(n1);

vector<int> R(n2);

// Copy data to temporary arrays L[] and R[]

for (int i = 0; i < n1; i++)

L[i] = arr[left + i];

for (int j = 0; j < n2; j++)

R[j] = arr[mid + 1 + j];

// Merge the temporary arrays back into arr[left..right]

int i = 0; // Initial index of first subarray

int j = 0; // Initial index of second subarray

int k = left; // Initial index of merged subarray

while (i < n1 && j < n2)

{

if (L[i] <= R[j])

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

// Copy the remaining elements of L[], if there are any

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

// Copy the remaining elements of R[], if there are any

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

void mergeSort(vector<int> &arr, int left, int right)

{

if (left < right)

{

int mid = left + (right - left) / 2;

// Sort first and second halves

mergeSort(arr, left, mid);

mergeSort(arr, mid + 1, right);

// Merge the sorted halves

merge(arr, left, mid, right);

}

}

int main()

{

vector<int> arr = {12, 11, 13, 5, 6, 7};

int n = arr.size();

cout << "Original array: ";

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

{

cout << arr[i] << " ";

}

cout << endl;

mergeSort(arr, 0, n - 1);

cout << "Sorted array: ";

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

{

cout << arr[i] << " ";

}

cout << endl;

return 0;

}

**OUTPUT**

****

1. **Write the above programs (1,2, and 3) using functions and call by address only.**

**Code-**

#include <iostream>

using namespace std;

void linearSearch(int \*arr, int n, int num){

int c =0;

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

{

if (arr[i] == num)

{

c++;

break;

}

}

if (c!=0)

{

cout << num << " is present in the array\n";

cout<<c;

}

else

{

cout << num << " is not present in the array\n";

}

}

// }

int main()

{

int n;

int num;

int i;

while (n != 0)

{

cout << "Enter the size of array:" << endl;

cin >> n;

if (n == 0)

{

cout << "program terminated\n"<< endl;

}

int \*arr = new int(n);

cout << "Enter all numbers:\n";

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

{

cin >> arr[i];

}

cout << "enter the no. to check that is present in array or not:" << endl;

cin >> num;

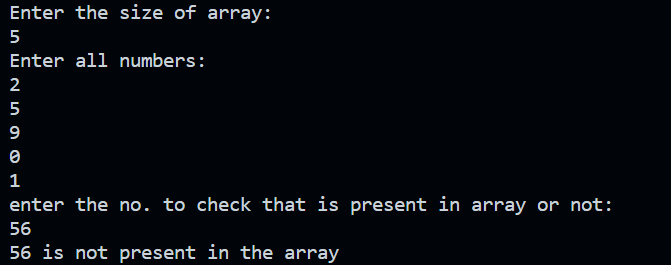
linearSearch(arr, n,num);

return 0;

}

}

OUTPUT



**6b)**

Code-

#include <bits/stdc++.h>

using namespace std;

void findSecondMaxAndMin(int \*arr, int n){

int firstMax=INT\_MIN, secondMax=INT\_MIN;

int firstMin=INT\_MAX, secondMin=INT\_MAX;

for(int i=0;i<n;i++){

if(arr[i]>firstMax){

secondMax=firstMax;

firstMax=arr[i];

}

else if (arr[i]>secondMax && arr[i]<firstMax){

secondMax=arr[i];

}

}

for(int i =0; i<n; i++){

if (arr[i]<firstMin){

secondMin=firstMin;

firstMin=arr[i];

}

else if(arr[i]<secondMin && arr[i]!=firstMin){

secondMin=arr[i];

}

}

cout<<"2nd maximum "<<secondMax<<"\n";

cout<<"2nd minimum "<<secondMin<<"\n";

}

int main(){

int n;

cout<<"Enter size of array;\n";

cin>>n;

int \*arr = new int[n];

cout<<"enter array elements:\n";

for(int i=0;i<n;i++){

cin>>arr[i];

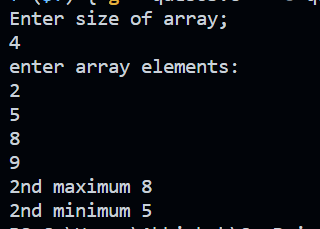
}

findSecondMaxAndMin(arr, n);

return 0;

}

**OUTPUT**



**6c)**

Code

#include <bits/stdc++.h>

using namespace std;

// question; insert element in array

// say 1,2,3 are elements and 4 is to be inserted at 2 position

// so new array becomes 1,2,4,3

void Change(int \*arr, int n)

{

int number, position;

cout << "enter number to be entered\n";

cin >> number;

cout << "Enter position where no. you want to add starting ftom 0\n";

cin >> position;

for (int i = n; i >= position; i--)

{

arr[i + 1] = arr[i];

}

arr[position] = number;

cout << "new array becomes:\n";

for (int i = 0; i < n + 1; i = i + 1)

{

cout << arr[i];

}

}

int main()

{

int n;

cout << "enter size of array(n)\n";

cin >> n;

int elements;

int \*arr = new int[n];

cout << "enter array elements:" << endl;

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

{

cin >> arr[i];

}

Change(arr, n);

return 0;

}

**OUTPUT**

