# Data Structures and Algorithms(UCS540)

Sixth-Semester

Submitted by:

Naman Sood [102104012] 3EE2

BE Third Year (2021-2025) Electrical Engineering

Submitted To: Dr. Yadvendra Singh

Assistant Professor

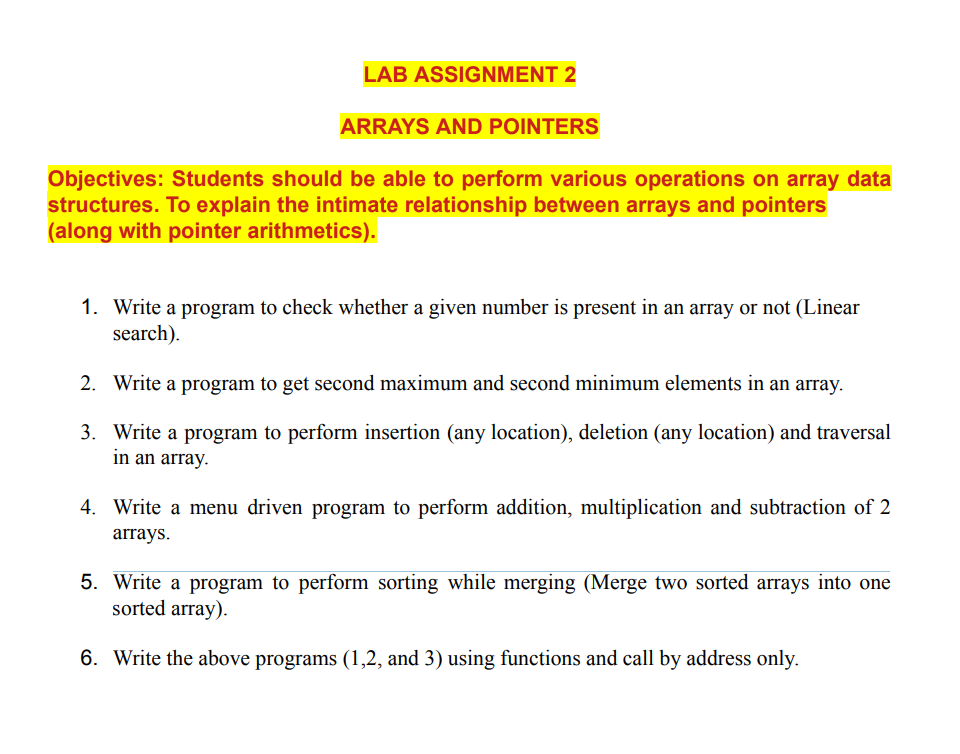


Department of Electrical & Instrumentation Engineering,

Thapar Institute of Engineering & Technology, Patiala

January-May 2024

# List of Experiments



**Q1.**

#include<iostream>

using namespace std;

void take\_input(int\* arr, int n)

{

cout<<"Input the elements"<<endl;

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

{

cin>>arr[i];

}

return;

}

int search(int\* arr, int n, int k)

{

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

{

if(arr[i] == k)

return i;

}

return -1;

}

int main()

{

int n;

cout<<"Enter the number of elements"<<endl;

cin>>n;

int\* arr = new int[n];

take\_input(arr,n);

int k;

cout<<"Enter the element to search for: "<<endl;

cin>>k;

int index = search(arr,n,k);

if(index != -1)

cout<<"The element "<<k<<" is found at position "<<index+1<<endl;

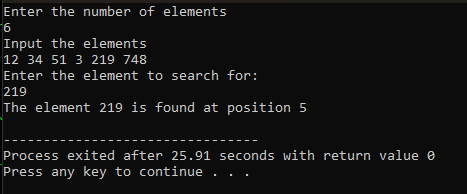
else

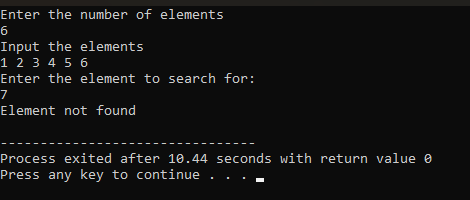
cout<<"Element not found"<<endl;

return 0;

}

**Output:**





**Q2.**

#include<iostream>

using namespace std;

void take\_input(int\* arr, int n)

{

cout<<"Input the elements"<<endl;

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

{

cin>>arr[i];

}

return;

}

int second\_min(int\* arr, int n)

{

int min = INT\_MAX,sec\_min = INT\_MAX;

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

{

if(arr[i]<min)

{

sec\_min = min;

min = arr[i];

}

if(arr[i]<sec\_min && arr[i]>min)

{

sec\_min = arr[i];

}

}

return sec\_min;

}

int second\_max(int\* arr, int n)

{

int max = INT\_MIN,sec\_max = INT\_MIN;

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

{

if(arr[i]>max)

{

sec\_max = max;

max = arr[i];

}

if(arr[i]>sec\_max && arr[i]<max)

{

sec\_max = arr[i];

}

}

return sec\_max;

}

int main()

{

int n;

cout<<"Enter the number of elements"<<endl;

cin>>n;

int\* arr = new int[n];

take\_input(arr,n);

int sec\_min = second\_min(arr,n);

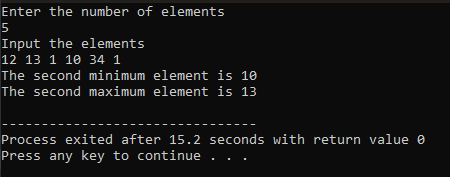
int sec\_max = second\_max(arr,n);

cout<<"The second minimum element is "<<sec\_min<<"\nThe second maximum element is "<<sec\_max<<endl;

return 0;

}

**Output:**



**Q3.**

#include <iostream>

const int MAX\_SIZE = 100;

void insertElement(int arr[], int& size) {

if (size >= MAX\_SIZE) {

std::cout << "Array is full. Cannot insert more elements." << std::endl;

return;

}

int position, element;

std::cout << "Enter the position to insert: ";

std::cin >> position;

if (position < 0 || position > size) {

std::cout << "Invalid position. Insertion failed." << std::endl;

return;

}

std::cout << "Enter the element to insert: ";

std::cin >> element;

for (int i = size; i > position; --i) {

arr[i] = arr[i - 1];

}

arr[position] = element;

++size;

std::cout << "Element inserted successfully." << std::endl;

}

void deleteElement(int arr[], int& size) {

if (size == 0) {

std::cout << "Array is empty. Cannot delete elements." << std::endl;

return;

}

int position;

std::cout << "Enter the position to delete: ";

std::cin >> position;

if (position < 0 || position >= size) {

std::cout << "Invalid position. Deletion failed." << std::endl;

return;

}

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

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

}

--size;

std::cout << "Element deleted successfully." << std::endl;

}

void traverseArray(const int arr[], int size) {

std::cout << "Array Elements: ";

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

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

}

std::cout << std::endl;

}

int main() {

int arr[MAX\_SIZE];

int size = 0;

int choice;

do {

std::cout << "\nMenu:\n";

std::cout << "1. Insert Element\n";

std::cout << "2. Delete Element\n";

std::cout << "3. Traverse Array\n";

std::cout << "4. Exit\n";

std::cout << "Enter your choice: ";

std::cin >> choice;

switch (choice) {

case 1:

insertElement(arr, size);

break;

case 2:

deleteElement(arr, size);

break;

case 3:

traverseArray(arr, size);

break;

case 4:

std::cout << "Exiting program.\n";

break;

default:

std::cout << "Invalid choice. Please try again.\n";

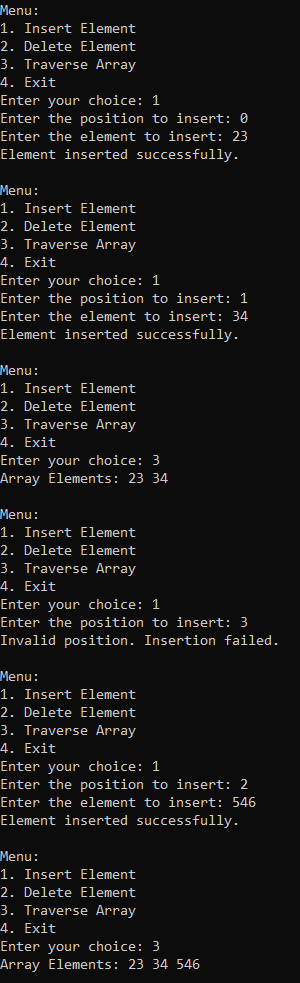
}

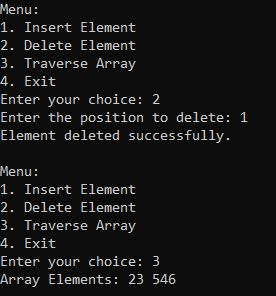
} while (choice != 4);

return 0;

}

**Output:**





**Q4.**

#include<iostream>

#include<cmath>

using namespace std;

void displayMenu()

{

cout<<"'1': Addition\n";

cout<<"'2': Subtraction\n";

cout<<"'3': Multiplication\n";

cout<<"'0': Exit\n";

}

int main()

{

int i,j;

int arr1[]={1,2,3,4,5};

int arr2[]={6,7,8,9,10};

int choice;

do

{

displayMenu();

cout<<"Enter Choice: ";

cin>>choice;

switch (choice)

{

case 0:

break;

case 1:

int sumarr[10];

cout<<"Addition of the two arrays:";

for(i=0;i<5;i++)

{

sumarr[i]=arr1[i]+arr2[i];

cout<<sumarr[i]<<" ";

}

cout<<endl;

break;

case 2:

int subarr[10];

cout<<"Subtraction of the two arrays:";

for(i=0;i<5;i++)

{

subarr[i] = abs(arr1[i]-arr2[i]);

cout<< subarr[i]<<" ";

}

cout<<endl;

break;

case 3:

int mularr[20];

cout<<"Multiplication of the two arrays:";

for(j=0;j<5;j++)

{

mularr[j]=arr1[j]\*arr2[j];

cout<< mularr[j]<<" ";

}

cout<<endl;

break;

default :

cout<<"Invalid Choice";

break;

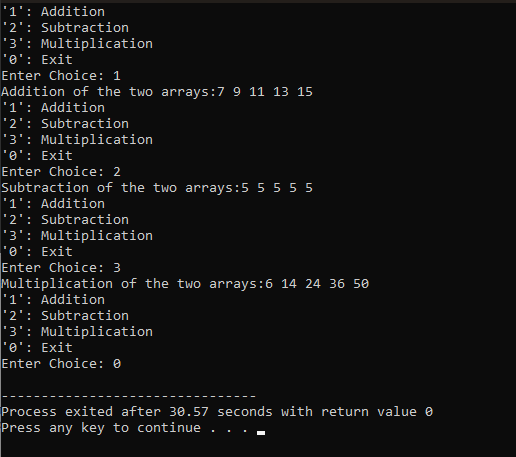
}

}while(choice!=0);

return 0;

}

**Output:**



**Q5.**

#include <iostream>

using namespace std;

void merge(int arr1[], int size1, int arr2[], int size2, int ans[])

{

int size = size1 + size2;

int k = 0;

int j = 0;

int i = 0;

while (i < size)

{

if (k < size1 && j < size2)

{

if (arr1[k] < arr2[j])

{

ans[i] = arr1[k];

k++;

}

else if (arr1[k] > arr2[j])

{

ans[i] = arr2[j];

j++;

}

else if (arr1[k] == arr2[j])

{

ans[i] = arr1[k];

k++;

}

}

else

{

if (k == size1)

{

ans[i] = arr2[j];

j++;

}

else if (j == size2)

{

ans[i] = arr1[k];

k++;

}

}

i++;

}

}

int main()

{

int size1;

cout<<"Input the size of the 1st array"<<endl;

cin >> size1;

int arr1[size1];

cout<<"Input elements of the 1st array"<<endl;

for (int i = 0; i < size1; i++)

{

cin >> arr1[i];

}

int size2;

cout<<"Input the size of the 2nd array"<<endl;

cin >> size2;

int arr2[size2];

cout<<"Input elements of the 2nd array"<<endl;

for (int i = 0; i < size2; i++)

{

cin >> arr2[i];

}

int ans[size1+size2];

merge(arr1, size1, arr2, size2, ans);

cout<<"Resultant Merged Sorted Array"<<endl;

for (int i = 0; i < size1 + size2; i++)

{

cout << ans[i] << " ";

}

cout << endl;

}

**Output:**

