**Chapter 10: Functions & Applications of Arrays**

**Laboratory Exercises (4)**

**Arrays and Loops**

**KEYWORDS: array, for, bool**

**Program 1:** Write a function to print Sum and average of all the elements of an array. The parameters to function are array and size of the array. Use the function in your program for array of 10 elements.

#include<iostream>

using namespace std;

void sumAverage(int B[], int size)

{

int sum=0, average;

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

sum=sum+B[i];

average=sum/size;

cout<<"Sum ="<<sum<<endl;

cout<<"Average = "<<average<<endl;

}

int main()

{

int A[10];

cout<<"Enter 10 Numbers :"<<endl;

for (int i=0; i<10; i++)

cin>>A[i];

sumAverage(A,10);

return 0;

}****

**Sample Output**

**Program 2:** Write a function to return the largest element of an array. The parameters to function are array and size of array. Use the function in your program for array of 10 elements.

#include<iostream>

using namespace std;

int largest(int B[], int size)

{

int large=B[0];

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

if(B[i]>large)

large=B[i];

return large;

}

int main()

{

int A[10];

cout<<"Enter 10 Numbers :"<<endl;

for (int i=0; i<10; i++)

cin>>A[i];

cout<<"Largest Number = "<<largest(A,10) <<endl;

return 0;

}****

**Sample Output**

**Program 3:** Write a function to reverse the positions of elements of an array. Thus, the first element becomes last element of the array. The parameters to function are array and size of array. Use the function in your program for array of 10 elements.

#include<iostream>

using namespace std;

void reverse(int B[], int size)

{

int j=size-1;

for(int i=0; i<(size/2);i++)

{

int temp=B[i];

B[i]=B[j];

B[j]=temp;

j--;

}

}

int main()

{

int A[10];

cout<<"Enter 10 Numbers :"<<endl;

for (int i=0; i<10; i++)

cin>>A[i];

reverse(A,10);

cout<<endl<<"Array in Reverse Order :"<<endl;

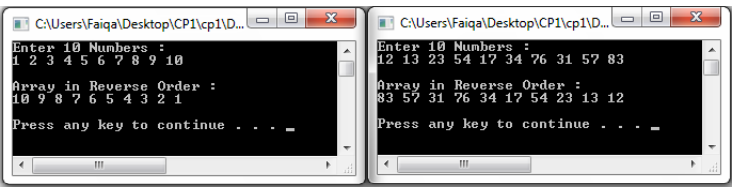
for (int i=0; i<10; i++)

cout<<A[i]<<" ";

cout<<endl<<endl;

return 0;

}

****

**Sample Output**

**2 Dimensional Arrays**

**KEYWORDS: array, for, nested loops.**

**Program 1:** Write a program for addition of 3x3 arrays using function. The function should take three arrays and number of rows and columns as arguments.

#include<iostream>

using namespace std;

void EnterArray(int X[][3])

{

for (int i=0; i<3;i++)

for (int j=0; j<3;j++)

cin>>X[i][j];

}

void PrintArray(int X[][3])

{

for (int i=0; i<3;i++)

{

for (int j=0; j<3;j++)

cout<<X[i][j]<<"\t";

cout<<endl;

}

cout<<endl;

}

void Addition(int Y[][3], int Z[][3])

{

for (int i=0; i<3;i++)

{

for (int j=0; j<3;j++)

cout<< Y[i][j] + Z[i][j]<<"\t";

cout<<endl;

}

}

int main()

{

int A[3][3], B[3][3];

cout<<"Enter the values for Matrix A:"<<endl;

EnterArray(A);

cout<<"Enter the values for Matrix B:"<<endl;

EnterArray(B);

cout<<endl<<"Matrix A:"<<endl;

PrintArray(A);

cout<<endl<<"Matrix B:"<<endl;

PrintArray(B);

cout<<endl<<"Matrix Addition A + B:"<<endl;

Addition (A,B);

return 0;

}

Another answer:

#include<iostream>

using namespace std;

void EnterArray(int X[][3])

{

for (int i=0; i<3;i++)

for (int j=0; j<3;j++)

cin>>X[i][j];

}

void PrintArray(int X[][3])

{

for (int i=0; i<3;i++)

{

for (int j=0; j<3;j++)

cout<<X[i][j]<<"\t";

cout<<endl;

}

cout<<endl;

}

void Addition(int Y[][3], int Z[][3], int W[][3])

{

for (int i=0; i<3;i++)

for (int j=0; j<3;j++)

W[i][j] = Y[i][j] + Z[i][j];

}

int main()

{

int A[3][3], B[3][3], C[3][3];

cout<<"Enter the values for Matrix A:"<<endl;

EnterArray(A);

cout<<"Enter the values for Matrix B:"<<endl;

EnterArray(B);

cout<<endl<<"Matrix A:"<<endl;

PrintArray(A);

cout<<endl<<"Matrix B:"<<endl;

PrintArray(B);

cout<<endl<<"Matrix Addition A + B:"<<endl;

Addition (A,B,C);

for (int i=0; i<3;i++)

{

for (int j=0; j<3;j++)

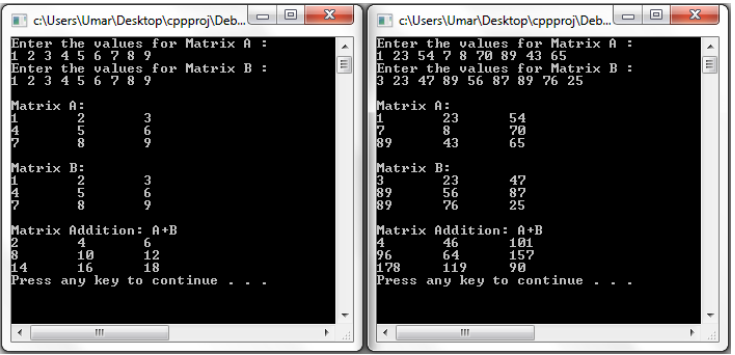
cout<<C[i][j]<<"\t";

cout<<endl;

}

return 0;

}



**Sample Output**

**Program 2:** Write a program for printing the largest elements in each row of an array using function. The function should take an array and number of rows and columns as parameters.

#include<iostream>

using namespace std;

void largestInRow(int r, int c, int X[][10])

{

int i, j, largest;

cout<<endl<<"Largest Values in each row of the Matrix :"<<endl;

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

{

largest=X[i][0];

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

{

if (largest < X[i][j])

{

largest = X[i][j];

}

}

cout<<"Row "<<i<<": "<<largest<<endl;

}

cout<<endl;

}

int main()

{

int A[10][10], r, c, i, j;

cout<<"Enter number of rows and columns of the Matrix (<= 10) :"<<endl;

cin>>r;

cin>>c;

cout<<endl<<"Enter "<<r\*c<<" values for the Matrix :"<<endl;

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

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

cin>>A[i][j];

cout<<endl<<"Matrix Values:"<<endl;

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

{

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

cout<<A[i][j]<<"\t";

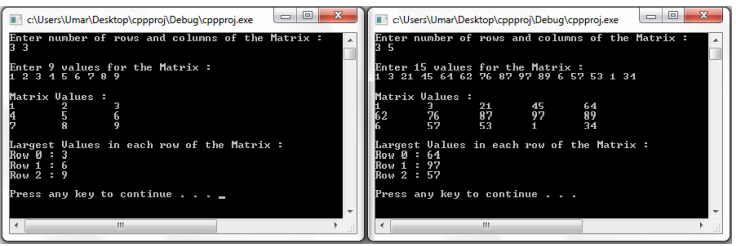
cout<<endl;

}

largestInRow(r,c,A);

return 0;

}

****

**Sample Output**

**Sorting and Searching Techniques using Arrays**

**KEYWORDS: array, for, nested loops.**

**Program 1:** Write a program for implementation of Sequential search using function.

See CH 10 slides

**Program 2:** Write a program for implementation of Binary search using function.

See CH 10 slides

A screenshot of a computer

Description automatically generated with medium confidence

**Sample Output**

**Program 3:** Write a program for implementation of Bubble sort using function.

See CH 10 slides

A screenshot of a computer

Description automatically generated with low confidence

**Sample Output**