

Data Structures & Algorithms Lab

Abdul Munim



September 14, 2024

Sap id: 55470

**Lab 03**

Write a program in C++ that takes integer type 2D array from user, calculate sum multiplication and average of all numbers.

#include <iostream>

using namespace std;

int main()

{

int w[2][2]{}, a[2][2]{}, sum, mul, avg;

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

{

for (int j = 0; j < 2; j++)

{

cout << "\n Enter element [" << i + 1 << "] Enter element [" << j + 1 << "] in the first array = ";

cin >> w[i][j];

}

}

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

{

for (int j = 0; j < 2; j++)

{

cout << "\n Enter element [" << i + 1 << "] Enter element [" << j + 1 << "] in the second array = ";

cin >> a[i][j];

}

}

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

{

for (int j = 0; j < 2; j++)

{

sum = w[i][j] + a[i][j];

mul = w[i][j] \* a[i][j];

avg = sum / 4;

}

}

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

cout << "\n Multiplication = " << mul << endl;

cout << "\n Average = " << avg << endl;

return 0;

}

**Output**

**A screenshot of a computer program

Description automatically generated**

Write a program in C++ to swap values of two variables using pointers.

#include <iostream>

using namespace std;

int main()

{

int a = 10, b = 20, temp = 0;

int\* ptrx = &a, \* ptry = &b;

cout << "\n Before Swapping a = " << a << " and b = " << b << endl;

temp = \*ptrx;

\*ptrx = \*ptry;

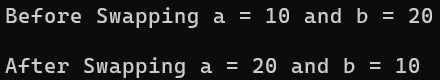
\*ptry = temp;

cout << "\n After Swapping a = " << a << " and b = " << b << endl;

return 0;

}

**Output**



Write a program that lets the user to enter the 10 values into the array. The program should then display the largest and the smallest value stored in the array.

#include <iostream>

using namespace std;

int main()

{

int a[10]{}, min, max = 0;

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

{

cout << "\n Enter elements [" << i + 1 << "] in the array = ";

cin >> a[i];

}

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

{

if (a[i] > max)

{

max = a[i];

}

}

min = a[0];

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

{

if (a[i] < min)

{

min = a[i];

}

}

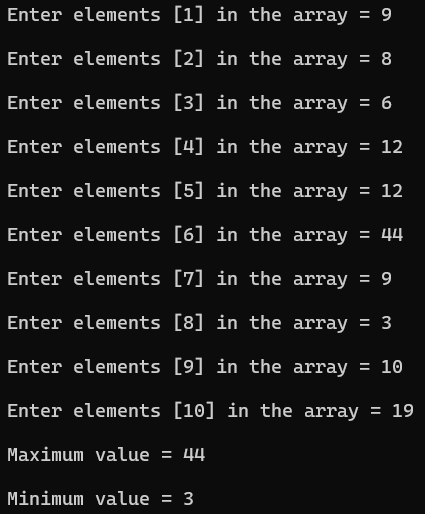
cout << "\n Maximum value = " << max << endl;

cout << "\n Minimum value = " << min << endl;

return 0;

}

**Output**

****

Write a program that lets the user to enter the total rainfall for each of 12 months into an array of doubles. The program should calculate and display the total rainfall of the year, the average monthly rainfall and the month with the highest and lowest rainfall.

#include <iostream>

using namespace std;

int main()

{

const int Months = 12;

double rainfall[Months]{}, totalRainfall = 0, averageRainfall, highestRainfall = 0, lowestRainfall = 0;

int highestMonth = 0, lowestMonth = 0;

cout << "\n Enter the total rainfall for each of the 12 Months: " << endl;

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

{

cout << " Month " << (i + 1) << " = ";

cin >> rainfall[i];

totalRainfall = totalRainfall + rainfall[i];

if (i == 0 || rainfall[i] > highestRainfall)

{

highestRainfall = rainfall[i];

highestMonth = i;

}

if (i == 0 || rainfall[i] < lowestRainfall)

{

lowestRainfall = rainfall[i];

lowestMonth = i;

}

}

averageRainfall = totalRainfall / Months;

cout << "\n Total rainfall for the year = " << totalRainfall << endl;

cout << " Average monthly rainfall = " << averageRainfall << endl;

cout << " Month with the Highest Rainfall = Month " << (highestMonth + 1) << " (" << highestRainfall <<")" << endl;

cout << " Month with the Lowest Rainfall = Month " << (lowestMonth + 1) << " (" << lowestRainfall <<")" << endl;

return 0;

}

**Output**

A screenshot of a computer

Description automatically generated

Write a program that creates a two-dimensional array initialized with test data.

Perform the following operations on the array data.

Get total: get total of all the elements in the array.

Get Average: calculate the average of all the values in the array.

Get row total: calculate the total of all the values in the specified row.

Get column total: calculate the total of all the values in the specified column.

Get Highest in row: find the highest value in the specified row.

Get Highest in Column: find the highest value in the specified column.

#include <iostream>

using namespace std;

int main()

{

const int Rows = 3;

const int Columns = 4;

int a[Rows][Columns] = {

{20, 40, 60, 80},

{100, 120, 140, 160},

{180, 200, 220, 240}

};

int total = 0;

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

{

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

{

total = total + a[i][j];

}

}

cout << "\n Total of all elements = " << total << endl;

double average = static\_cast<double>(total) / (Rows \* Columns);//converting total, rows and columns in datatype double

cout << "\n Average of all elements = " << average << endl;

int row = 1;

int rowTotal = 0;

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

{

rowTotal = rowTotal + a[row][j];

}

cout << "\n Total of elements in row " << row << " = " << rowTotal << endl;

int col = 2;

int colTotal = 0;

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

{

colTotal = colTotal + a[i][col];

}

cout << "\n Total of elements in column " << col << " = " << colTotal << endl;

int HighestInRow = a[row][0];

for (int j = 1; j < Columns; j++)

{

if (a[row][j] > HighestInRow)

{

HighestInRow = a[row][j];

}

}

cout << "\n Highest value in row " << row << " = " << HighestInRow << endl;

int HighestInColumn = a[0][col];

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

{

if (a[i][col] > HighestInColumn)

{

HighestInColumn = a[i][col];

}

}

cout << "\n Highest value in column " << col << " = " << HighestInColumn << endl;

return 0;

}

**Output**

**A black background with white text

Description automatically generated**

Write a program that dynamically allocates an array of integers. Read the values from user and calculate the sum of odd integers.

#include <iostream>

using namespace std;

int main()

{

int size;

cout << "\n Enter the number of integers = ";

cin >> size;

int\* o = new int[size];

cout << " Enter " << size << " integers: " << endl;

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

{

cout << " ";

cin >> o[i];

}

int OddSum = 0;

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

{

if (o[i] % 2 != 0)

{

OddSum += o[i];

}

}

cout << "\n Sum of odd integers = " << OddSum << endl;

delete[] o;

return 0;

}

**Output**

**A black screen with white text

Description automatically generated**

Define a pointer variable. Assign the address of variable to a pointer variable and access the value of address variable in the pointer variable.

In C++, pointers are variables that store the memory addresses of other variables.

#include <iostream>

using namespace std;

int main()

{

int num = 4;

int\* ptr = &num;

cout << "\n The value of num = " << num << endl;

cout << " The address of num = " << &num << endl;

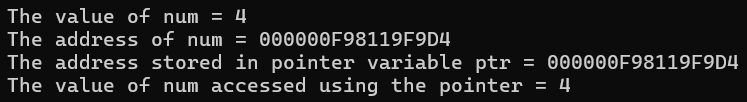
cout << " The address stored in pointer variable ptr = " << ptr << endl;

cout << " The value of num accessed using the pointer = " << \*ptr << endl;

return 0;

}

**Output**



Write a program that asks the user to enter integers as inputs to be stored in the variables 'a' and 'b' respectively. There are also two integer pointers named ptrA and ptrB. Assign the values of 'a' and 'b' to ptrA and ptrB respectively and display them.

#include <iostream>

using namespace std;

int main()

{

int a, b;

int\* ptrA;

int\* ptrB;

cout << "\n Enter the value for a = ";

cin >> a;

cout << " Enter the value for b = ";

cin >> b;

ptrA = &a;

ptrB = &b;

cout << "\n The value of a is = " << \*ptrA << endl;

cout << " The value of b is = " << \*ptrB << endl;

return 0;

}

**Output**

A black background with white text

Description automatically generated

Write a program for a calculator using functions.

Your program must have the following functions:

a. Menu () telling the user to select from the options

b. Addition (int a, int b) adding two numbers

c. Subtraction (int a, int b)

d. Division (int a, int b)

e. Multiplication (int a, int b)

f. Pow (int number, int pow)

In the main function you will only call the menu () function.

#include <iostream>

#include <cmath>

using namespace std;

void Menu();

int Addition(int a, int b);

int Subtraction(int a, int b);

int Multiplication(int a, int b);

double Division(int a, int b);

int Pow(int number, int power);

void Menu()

{

int choice;

int num1, num2, result;

do {

cout << "\n Calculator Menu:" << endl;

cout << " 1. Addition" << endl;

cout << " 2. Subtraction" << endl;

cout << " 3. Multiplication" << endl;

cout << " 4. Division" << endl;

cout << " 5. Power" << endl;

cout << " 6. Exit" << endl;

cout << " Enter your choice (1-6) = ";

cin >> choice;

switch (choice)

{

case 1:

cout << "\n Enter first integer = ";

cin >> num1;

cout << " Enter second integer = ";

cin >> num2;

result = Addition(num1, num2);

cout << "Result = " << result << endl;

break;

case 2:

cout << "\n Enter first integer = ";

cin >> num1;

cout << " Enter second integer = ";

cin >> num2;

result = Subtraction(num1, num2);

cout << " Result = " << result << endl;

break;

case 3:

cout << "\n Enter first integer =: ";

cin >> num1;

cout << "\n Enter second integer = ";

cin >> num2;

result = Multiplication(num1, num2);

cout << " Result = " << result << endl;

break;

case 4:

cout << "\n Enter first integer = ";

cin >> num1;

cout << "\n Enter second integer = ";

cin >> num2;

if (num2 == 0)

{

cout << " Error! Division by zero not possible" << endl;

}

else

{

double divResult = Division(num1, num2);

cout << " Result = " << divResult << endl;

}

break;

case 5:

cout << "\n Enter the base number = ";

cin >> num1;

cout << " Enter the power = ";

cin >> num2;

result = Pow(num1, num2);

cout << " Result = " << result << endl;

break;

case 6:

cout << "\n Exiting the calculator..." << endl;

break;

default:

cout << "\n Invalid choice! Please try again." << endl;

}

} while (choice != 6);

}

int Addition(int a, int b)

{

return a + b;

}

int Subtraction(int a, int b)

{

return a - b;

}

int Multiplication(int a, int b)

{

return a \* b;

}

double Division(int a, int b)

{

return static\_cast<double>(a) / b;

}

int Pow(int number, int power)

{

return static\_cast<int>(pow(number, power));

}

int main()

{

Menu();

return 0;

}

**Output**

A screenshot of a computer

Description automatically generated