**35. Program to reverse the elements in an array**

#include <iostream>

using namespace std;

int main() {

const int size = 5;

int arr[size];

cout << "Enter " << size << " elements for the array:" << endl;

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

cin >> arr[i];

}

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

int temp = arr[i];

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

arr[size - i - 1] = temp;

}

cout << "Reversed array:" << endl;

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

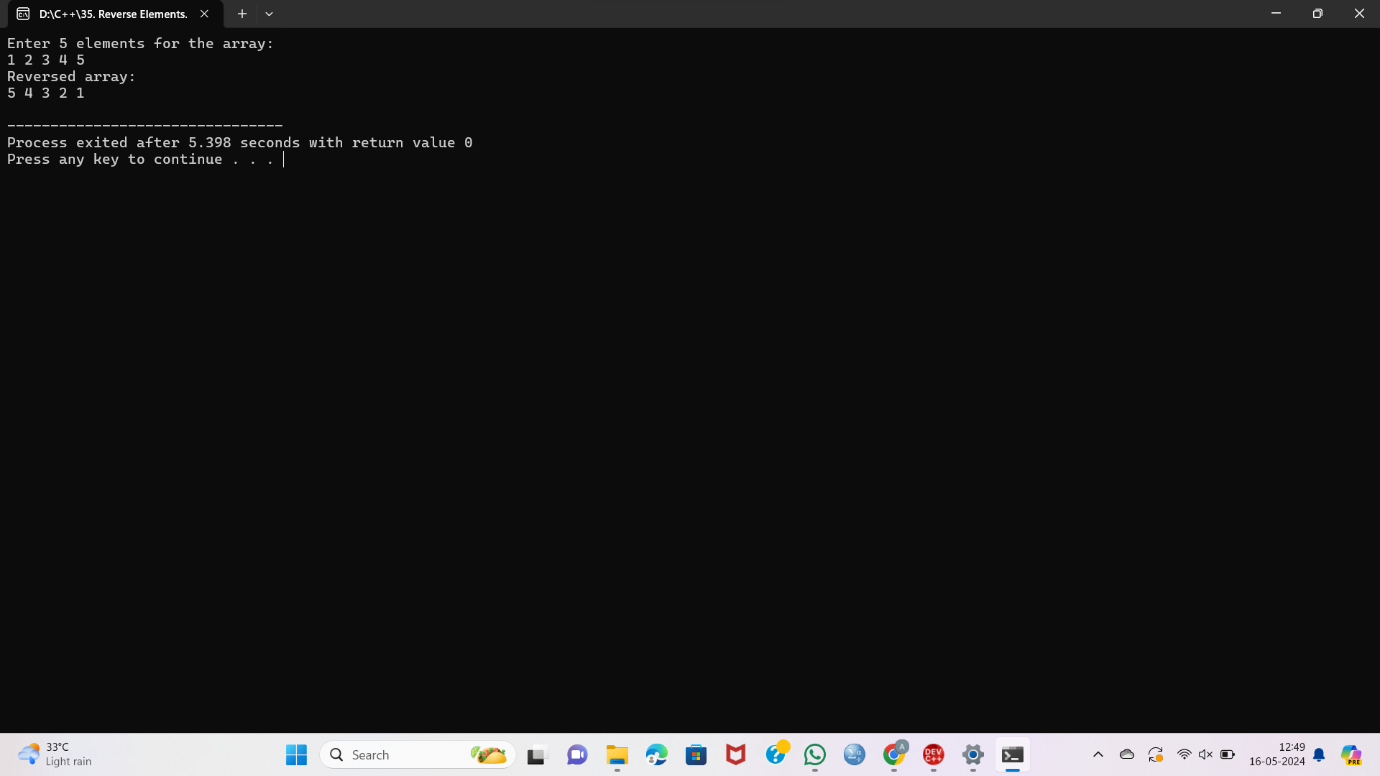
cout << arr[i] << " ";

}

cout << endl;

return 0;

}



**36. Program to insert an element in an array at a specific position**

#include <iostream>

using namespace std;

int main() {

const int max\_size = 100;

int arr[max\_size], size, element, position;

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

cin >> size;

cout << "Enter " << size << " elements for the array:" << endl;

for (int i = 0; i < size; i++) cin >> arr[i];

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

cin >> element;

cout << "Enter the position to insert (1 to " << size + 1 << "): ";

cin >> position;

if (position < 1 || position > size + 1) {

cout << "Invalid position!" << endl;

return 1;

}

for (int i = size; i >= position; i--) arr[i] = arr[i - 1];

arr[position - 1] = element;

size++;

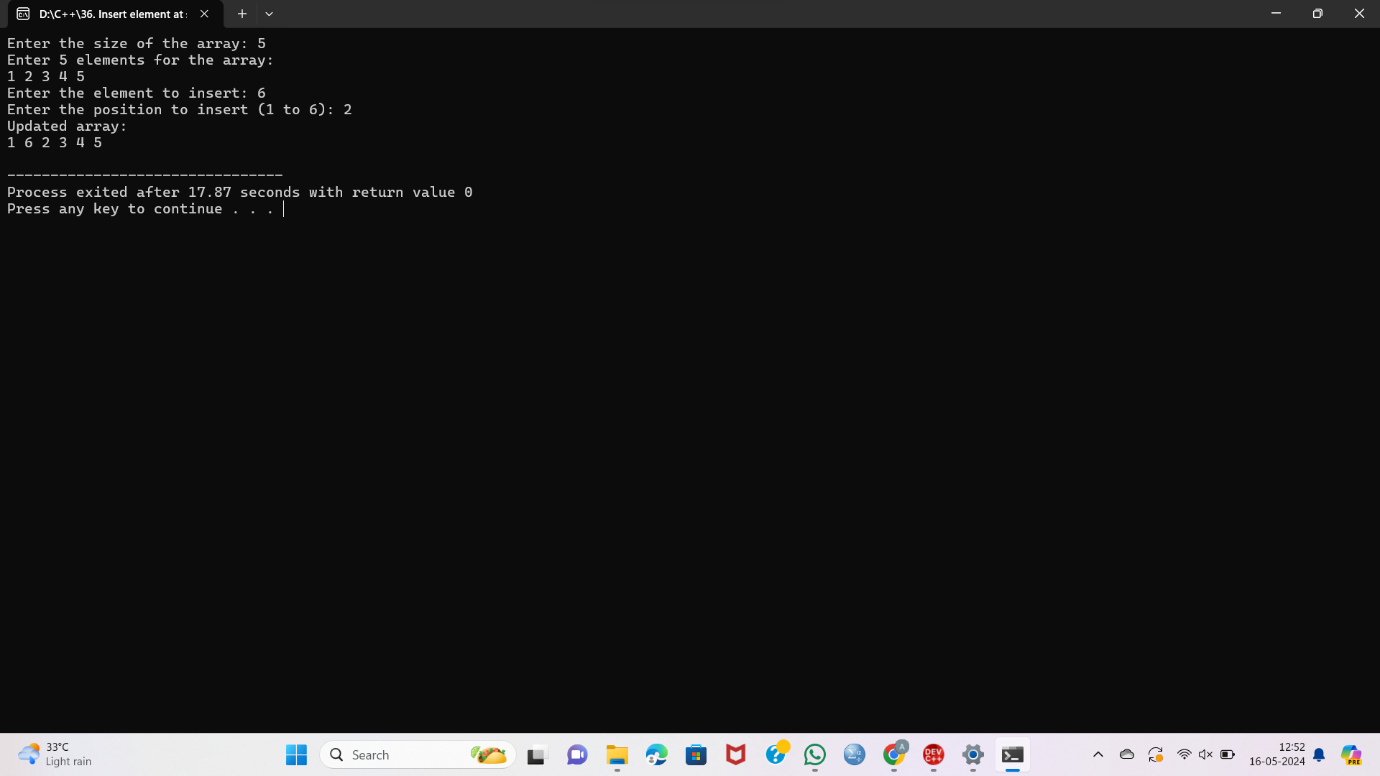
cout << "Updated array:" << endl;

for (int i = 0; i < size; i++) cout << arr[i] << " ";

cout << endl;

return 0;

}



**37. Program to Delete an element in an array at a specific position**

#include <iostream>

using namespace std;

int main() {

const int max\_size = 100;

int arr[max\_size];

int size, position;

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

cin >> size;

cout << "Enter " << size << " elements for the array:" << endl;

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

cin >> arr[i];

}

cout << "Enter the position to delete (1 to " << size << "): ";

cin >> position;

if (position < 1 || position > size) {

cout << "Invalid position!" << endl;

return 1;

}

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

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

}

size--;

cout << "Updated array:" << endl;

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

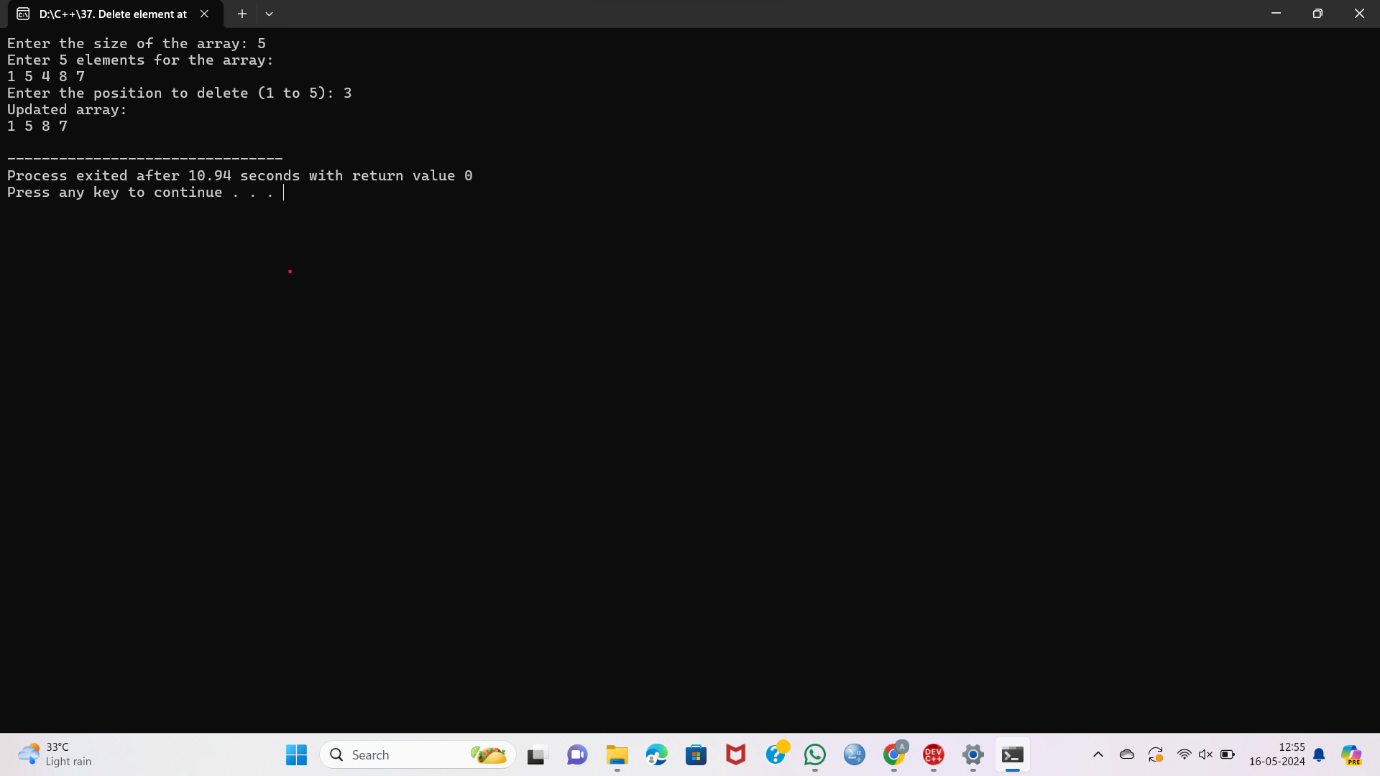
cout << arr[i] << " ";

}

cout << endl;

return 0;

}



**38. Find the sum of all elements in an array**

#include <iostream>

using namespace std;

int main() {

const int max\_size = 100;

int arr[max\_size];

int size, sum = 0;

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

cin >> size;

cout << "Enter " << size << " elements for the array:" << endl;

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

cin >> arr[i];

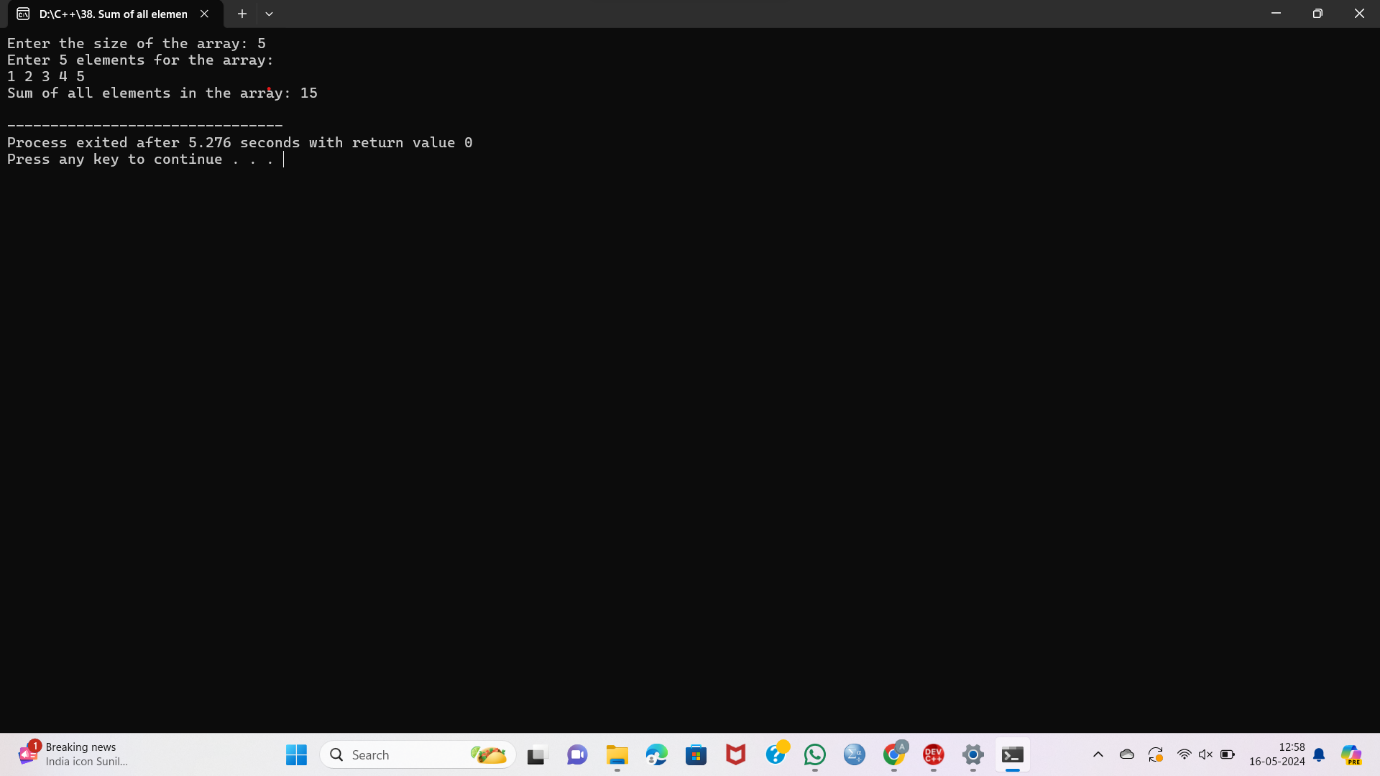
sum += arr[i];

}

cout << "Sum of all elements in the array: " << sum << endl;

return 0;

}



**39. Find the average of all elements in an array**

#include <iostream>

using namespace std;

int main() {

const int max\_size = 100;

int arr[max\_size];

int size;

double sum = 0, average;

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

cin >> size;

cout << "Enter " << size << " elements for the array:" << endl;

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

cin >> arr[i];

sum += arr[i];

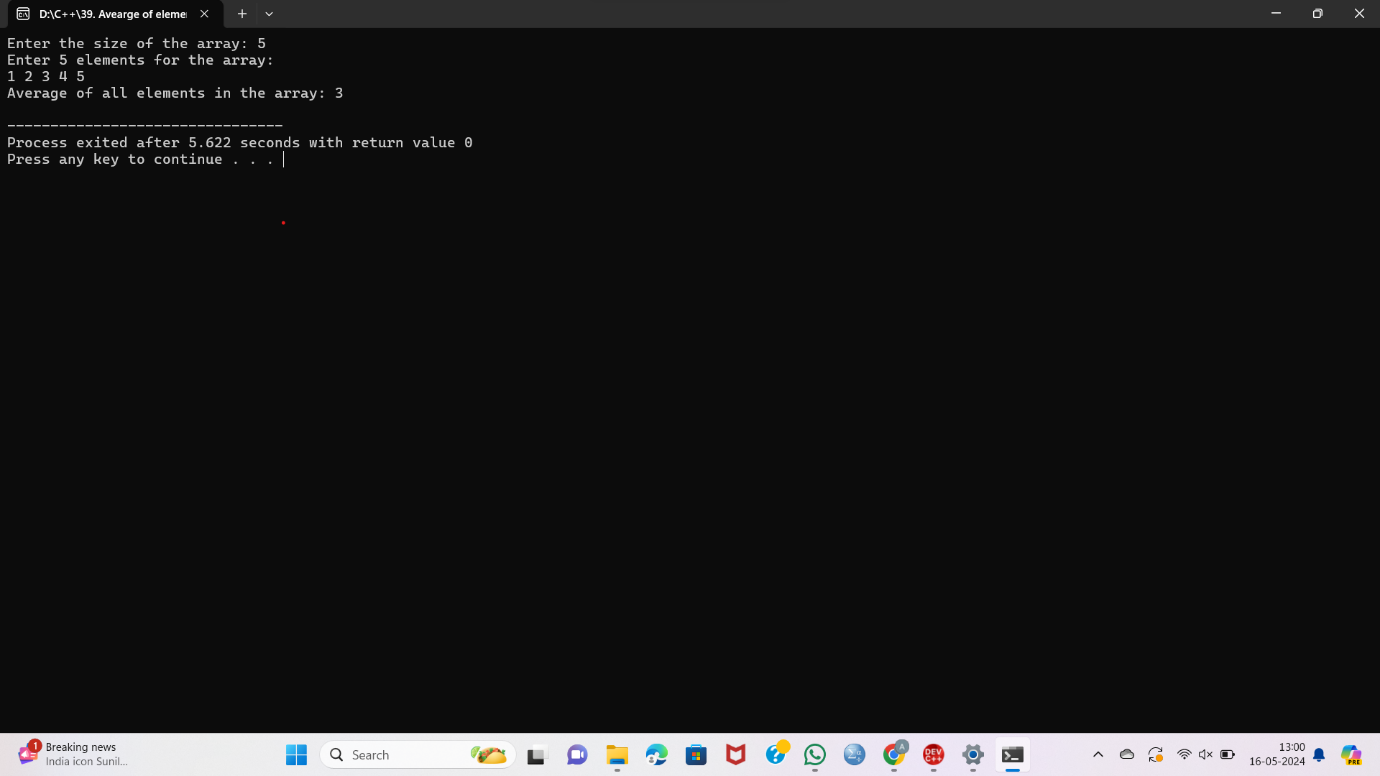
}

average = sum / size;

cout << "Average of all elements in the array: " << average << endl;

return 0;

}



**40. Find the second largest element in an array**

#include <iostream>

#include <climits>

using namespace std;

int main() {

const int max\_size = 100;

int arr[max\_size];

int size, largest, secondLargest;

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

cin >> size;

cout << "Enter " << size << " elements for the array:" << endl;

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

cin >> arr[i];

}

largest = secondLargest = INT\_MIN;

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

if (arr[i] > largest) {

secondLargest = largest;

largest = arr[i];

} else if (arr[i] > secondLargest && arr[i] != largest) {

secondLargest = arr[i];

}

}

if (secondLargest != INT\_MIN) {

cout << "Second largest element in the array: " << secondLargest << endl;

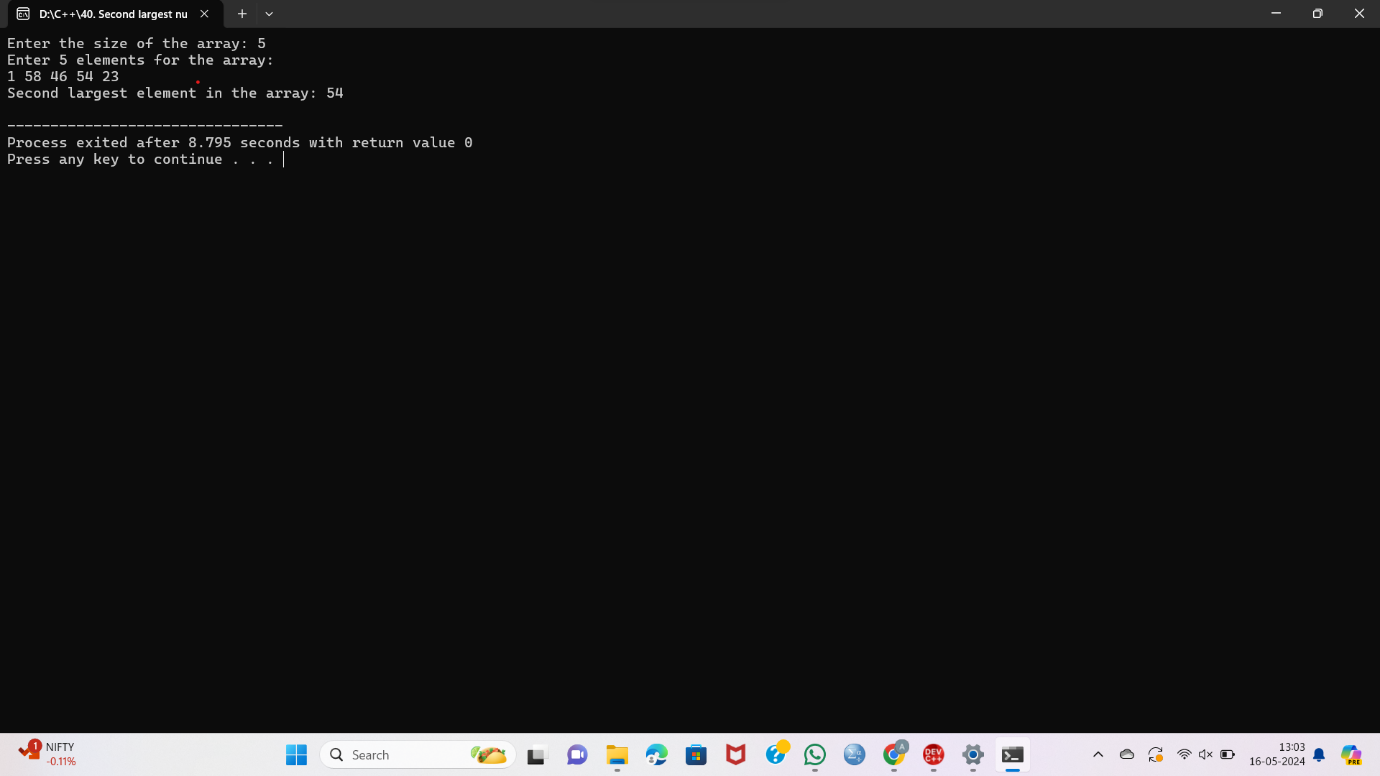
} else {

cout << "No second largest element found." << endl;

}

return 0;

}



**41. Find the number of occurrences of a value in an array**

#include <iostream>

using namespace std;

int main() {

const int max\_size = 100;

int arr[max\_size];

int size, value, count = 0;

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

cin >> size;

cout << "Enter " << size << " elements for the array:" << endl;

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

cin >> arr[i];

}

cout << "Enter the value to find occurrences: ";

cin >> value;

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

if (arr[i] == value) {

count++;

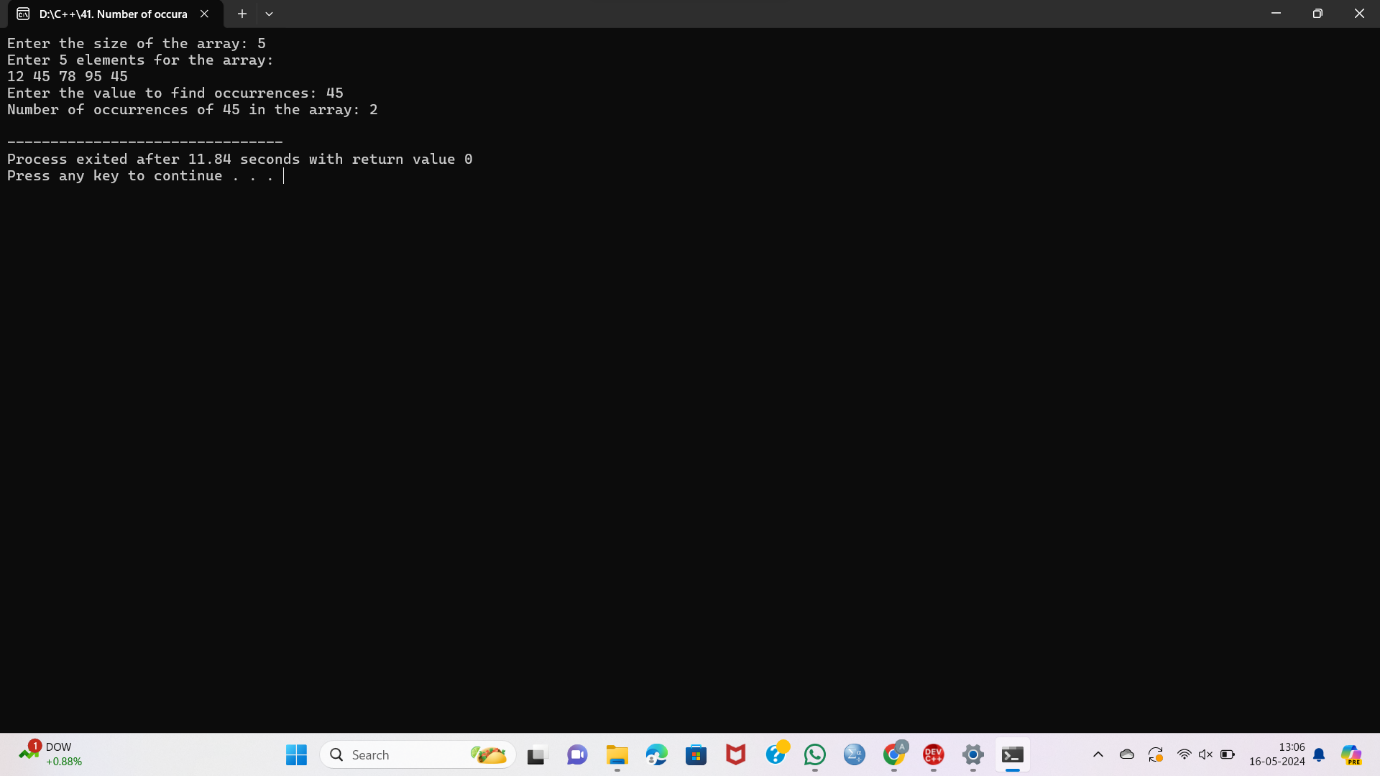
}

}

cout << "Number of occurrences of " << value << " in the array: " << count << endl;

return 0;

}



**42. Merge two array**

#include <iostream>

using namespace std;

int main() {

const int max\_size = 100;

int arr1[max\_size], arr2[max\_size], merged[max\_size \* 2];

int size1, size2, mergedSize;

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

cin >> size1;

cout << "Enter " << size1 << " elements for the first array:" << endl;

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

cin >> arr1[i];

}

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

cin >> size2;

cout << "Enter " << size2 << " elements for the second array:" << endl;

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

cin >> arr2[i];

}

mergedSize = size1 + size2;

int idx = 0;

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

merged[idx++] = arr1[i];

}

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

merged[idx++] = arr2[i];

}

cout << "Merged array:" << endl;

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

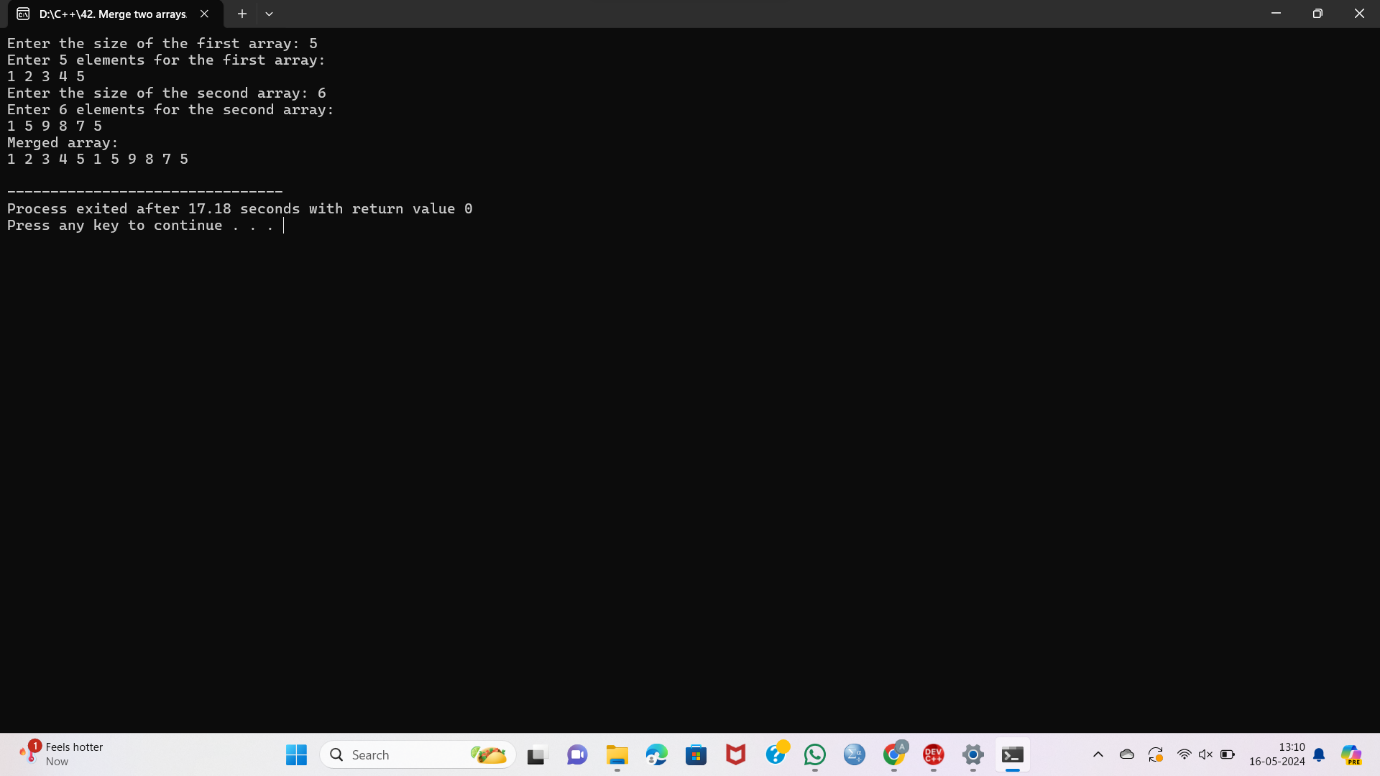
cout << merged[i] << " ";

}

cout << endl;

return 0;

}



**43. Create a dynamic array using pointers and display the values**

#include <iostream>

int main() {

int size;

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

std::cin >> size;

int\* arr = new int[size];

std::cout << "Enter " << size << " elements:\n";

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

std::cin >> arr[i];

}

std::cout << "Values in the array:\n";

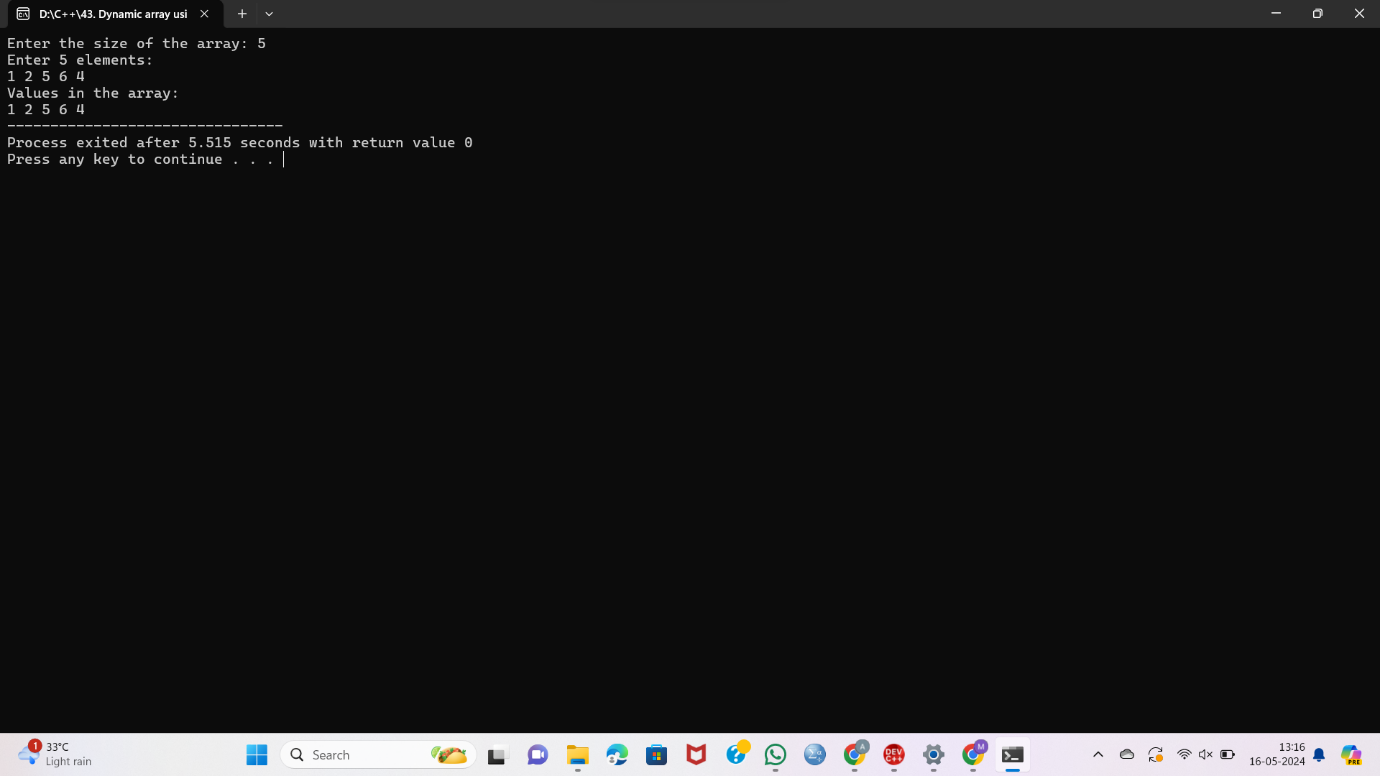
for (int i = 0; i < size; ++i) {

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

}

delete[] arr;

return 0;

}

**44. Add 2 matrices**

#include <iostream>

using namespace std;

int main() {

int rows, cols;

cout << "Enter the number of rows and columns for the matrices: ";

cin >> rows >> cols;

int matrix1[rows][cols], matrix2[rows][cols], sum[rows][cols];

cout << "Enter elements of the first matrix:\n";

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

for (int j = 0; j < cols; ++j) {

cin >> matrix1[i][j];

}

}

cout << "Enter elements of the second matrix:\n";

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

for (int j = 0; j < cols; ++j) {

cin >> matrix2[i][j];

}

}

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

for (int j = 0; j < cols; ++j) {

sum[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

cout << "Sum of the matrices:\n";

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

for (int j = 0; j < cols; ++j) {

cout << sum[i][j] << " ";

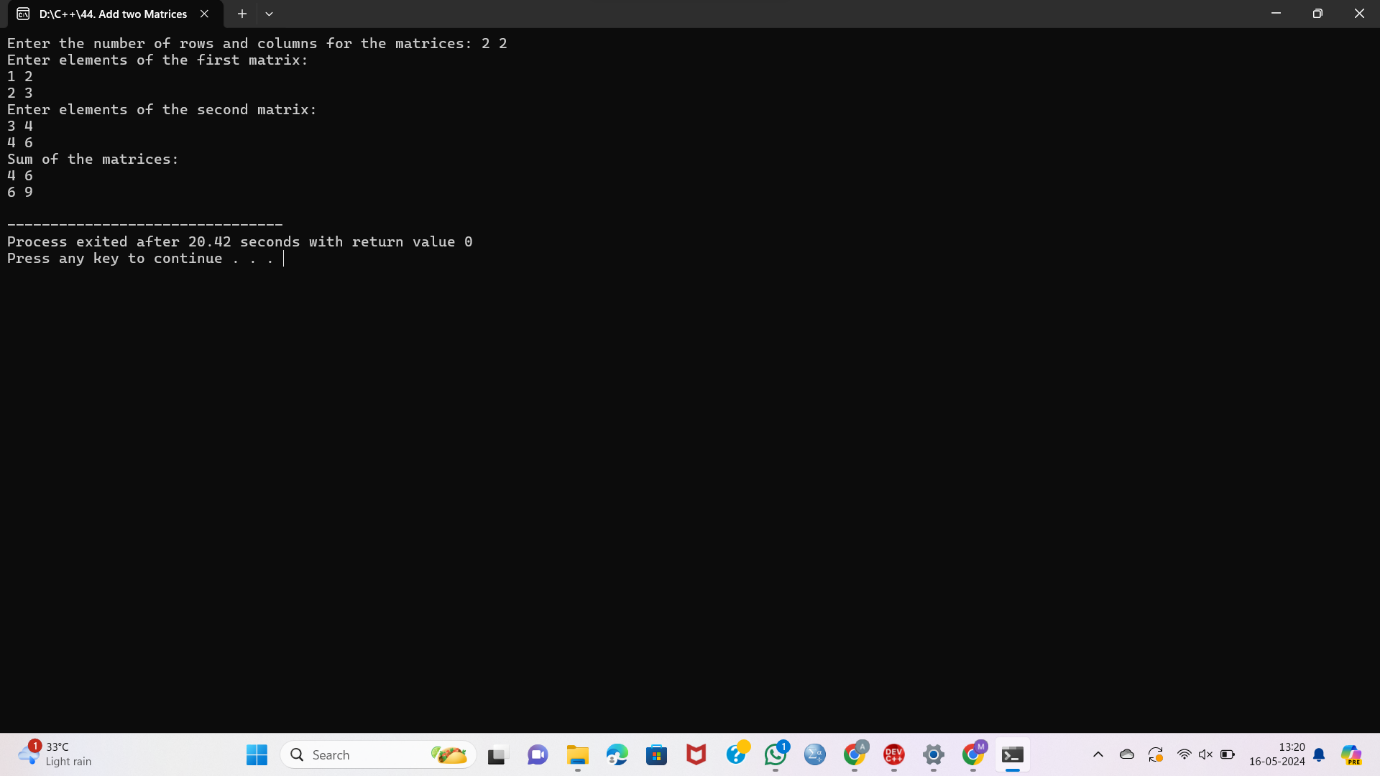
}

cout << endl;

}

return 0;

}



**45. Multiply 2 matrices**

#include <iostream>

using namespace std;

int main() {

int rows1, cols1, rows2, cols2;

cout << "Enter the number of rows and columns for the first matrix: ";

cin >> rows1 >> cols1;

cout << "Enter the number of rows and columns for the second matrix: ";

cin >> rows2 >> cols2;

if (cols1 != rows2) {

cout << "Matrix multiplication not possible. Columns of first matrix must be equal to rows of second matrix.";

return 0;

}

int matrix1[rows1][cols1], matrix2[rows2][cols2], result[rows1][cols2];

cout << "Enter elements of the first matrix:\n";

for (int i = 0; i < rows1; ++i) {

for (int j = 0; j < cols1; ++j) {

cin >> matrix1[i][j];

}

}

cout << "Enter elements of the second matrix:\n";

for (int i = 0; i < rows2; ++i) {

for (int j = 0; j < cols2; ++j) {

cin >> matrix2[i][j];

}

}

for (int i = 0; i < rows1; ++i) {

for (int j = 0; j < cols2; ++j) {

result[i][j] = 0;

}

}

for (int i = 0; i < rows1; ++i) {

for (int j = 0; j < cols2; ++j) {

for (int k = 0; k < cols1; ++k) {

result[i][j] += matrix1[i][k] \* matrix2[k][j];

}

}

}

cout << "Result of matrix multiplication:\n";

for (int i = 0; i < rows1; ++i) {

for (int j = 0; j < cols2; ++j) {

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

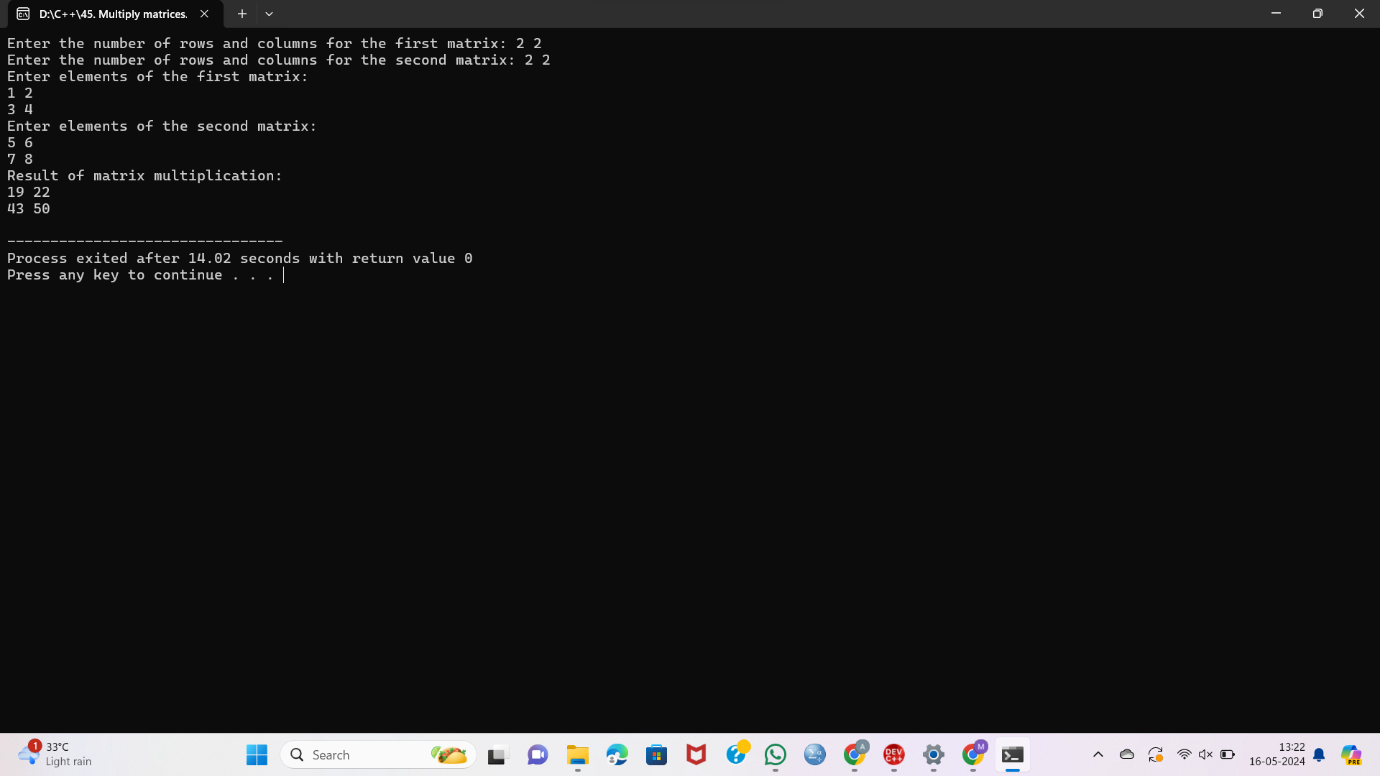
}

cout << endl;

}

return 0;

}



**46. Find the sum of diagonals of a matrix**

#include <iostream>

using namespace std;

int main() {

int rows, cols;

cout << "Enter the number of rows and columns for the matrix: ";

cin >> rows >> cols;

int matrix[rows][cols];

cout << "Enter elements of the matrix:\n";

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

for (int j = 0; j < cols; ++j) {

cin >> matrix[i][j];

}

}

int mainDiagonalSum = 0;

for (int i = 0; i < min(rows, cols); ++i) {

mainDiagonalSum += matrix[i][i];

}

int secondaryDiagonalSum = 0;

for (int i = 0; i < min(rows, cols); ++i) {

secondaryDiagonalSum += matrix[i][cols - i - 1];

}

cout << "Sum of main diagonal: " << mainDiagonalSum << endl;

cout << "Sum of secondary diagonal: " << secondaryDiagonalSum << endl;

return 0;

}

