35.Program to reverse the elements in an array.

Program:

#include <iostream>

using namespace std;

int main() {

int n;

// Take the number of elements in the array from the user

cout << "Enter the number of elements in the array: ";

cin >> n;

int arr[n];

// Take array elements input from the user

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

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

cin >> arr[i];

}

// Print original array

cout << "Original array: ";

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

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

}

cout << endl;

// Reverse the array in place

int start = 0;

int end = n - 1;

while (start < end) {

// Swap the elements

int temp = arr[start];

arr[start] = arr[end];

arr[end] = temp;

// Move to the next set of elements

start++;

end--;

}

// Print reversed array

cout << "Reversed array: ";

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

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

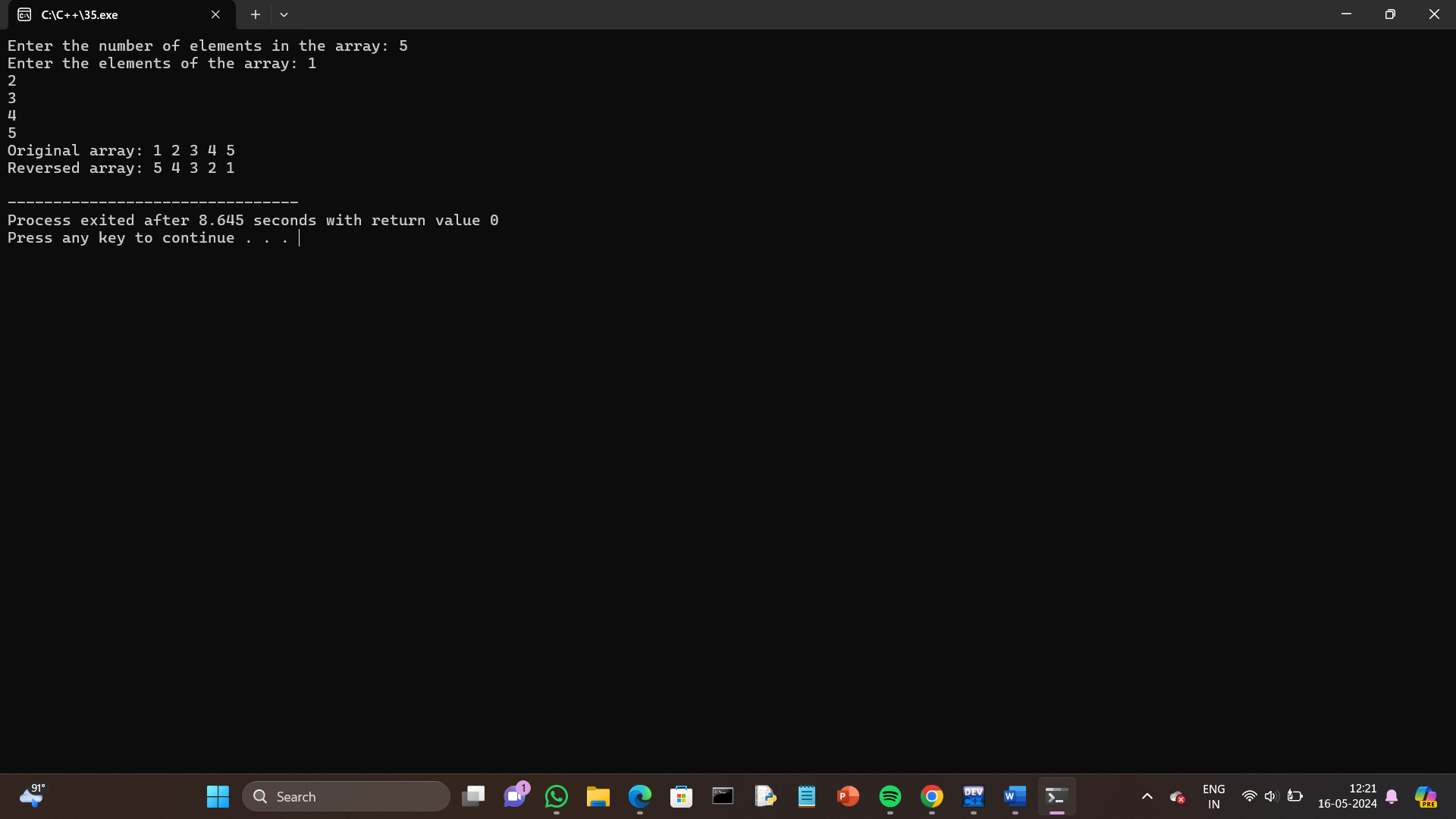
}

cout << endl;

return 0;

}

Output:



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

Program:

#include <iostream>

using namespace std;

int main() {

int n, pos, element;

// Take the number of elements in the array from the user

cout << "Enter the number of elements in the array: ";

cin >> n;

int arr[n + 1]; // Array size is n+1 to accommodate the new element

// Take array elements input from the user

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

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

cin >> arr[i];

}

// Take the position where the new element is to be inserted

cout << "Enter the position where the element should be inserted (0 to " << n << "): ";

cin >> pos;

// Validate position

if (pos < 0 || pos > n) {

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

return 1;

}

// Take the new element to be inserted

cout << "Enter the element to be inserted: ";

cin >> element;

// Shift elements to the right to make space for the new element

for (int i = n; i > pos; i--) {

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

}

// Insert the new element at the specified position

arr[pos] = element;

// Increment the size of the array

n++;

// Print the array after insertion

cout << "Array after insertion: ";

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

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

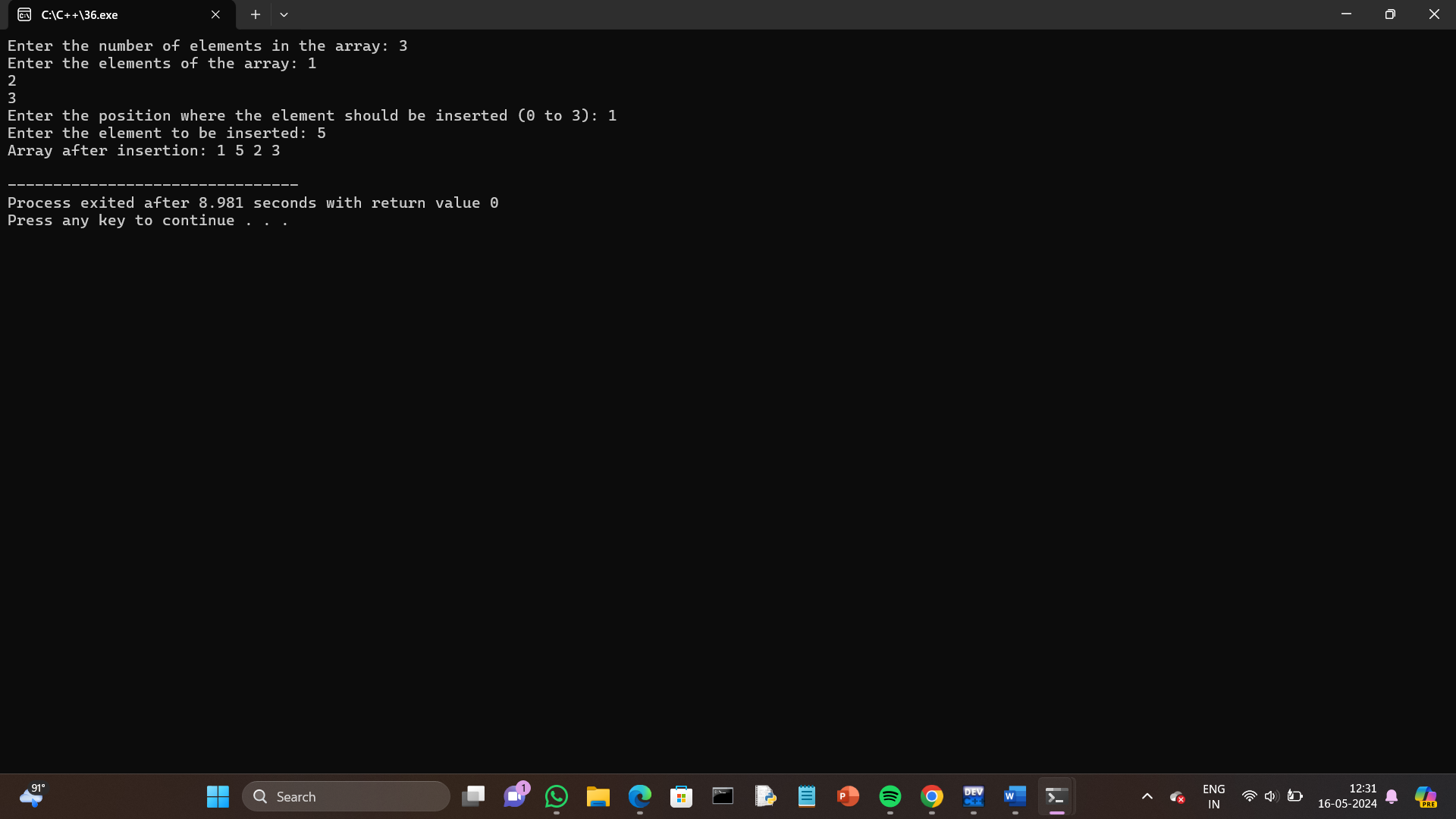
}

cout << endl;

return 0;

}

Output:



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

Program:

#include <iostream>

using namespace std;

int main() {

int n, pos;

// Take the number of elements in the array from the user

cout << "Enter the number of elements in the array: ";

cin >> n;

int arr[n]; // Initialize array with the given size

// Take array elements input from the user

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

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

cin >> arr[i];

}

// Take the position of the element to be deleted

cout << "Enter the position of the element to be deleted (0 to " << n - 1 << "): ";

cin >> pos;

// Validate position

if (pos < 0 || pos >= n) {

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

return 1;

}

// Shift elements to the left to delete the element at the specified position

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

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

}

// Decrement the size of the array

n--;

// Print the array after deletion

cout << "Array after deletion: ";

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

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

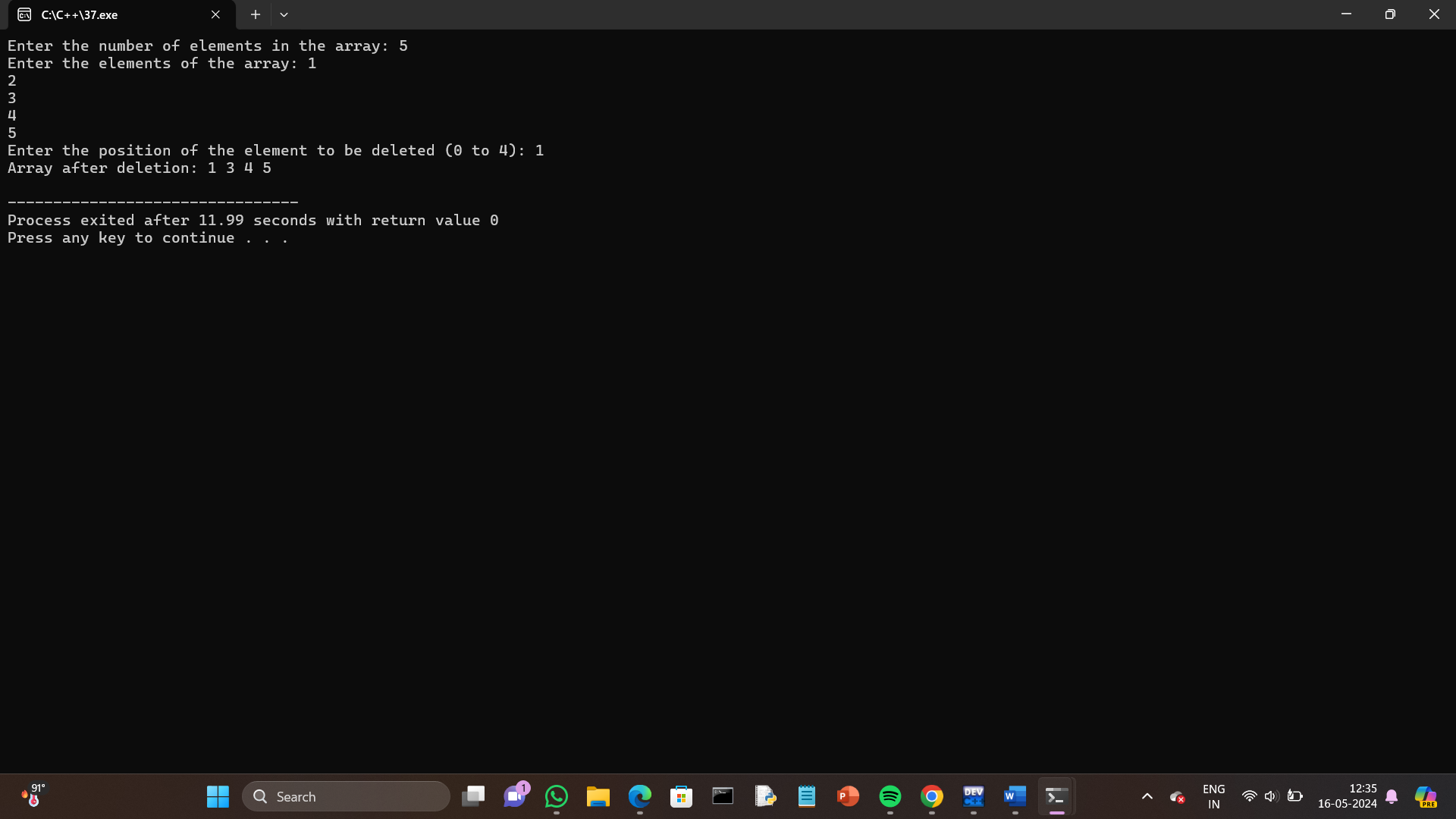
}

cout << endl;

return 0;

}

Output:



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

Program:

#include <iostream>

using namespace std;

int main() {

int n;

// Take the number of elements in the array from the user

cout << "Enter the number of elements in the array: ";

cin >> n;

int arr[n];

// Take array elements input from the user

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

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

cin >> arr[i];

}

// Calculate the sum of all elements in the array

int sum = 0;

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

sum += arr[i];

}

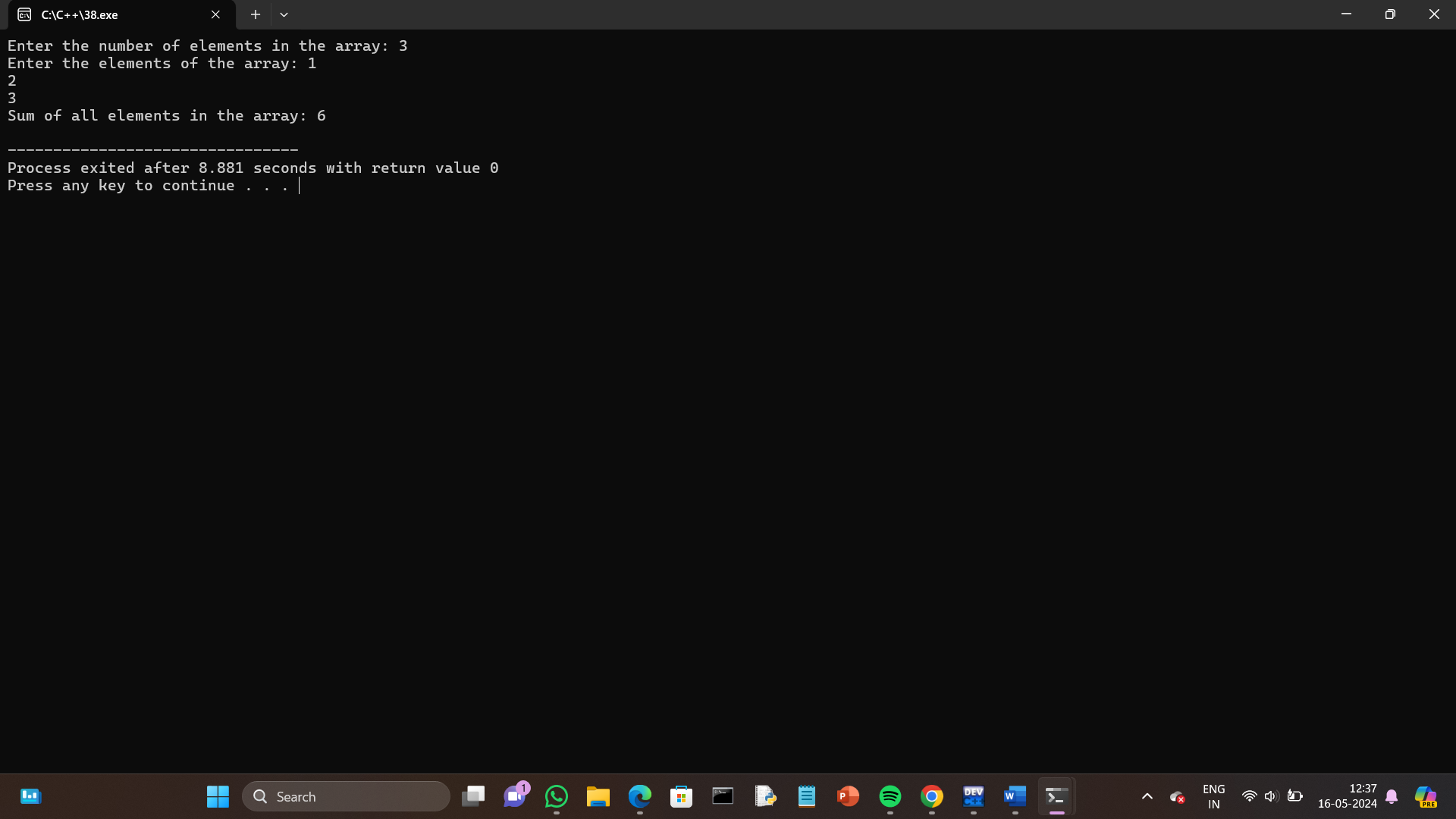
// Print the sum of the elements

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

return 0;

}

Output:



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

Program:

#include <iostream>

using namespace std;

int main() {

int n;

// Take the number of elements in the array from the user

cout << "Enter the number of elements in the array: ";

cin >> n;

int arr[n];

// Take array elements input from the user

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

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

cin >> arr[i];

}

// Calculate the sum of all elements in the array

int sum = 0;

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

sum += arr[i];

}

// Calculate the average

double average = static\_cast<double>(sum) / n;

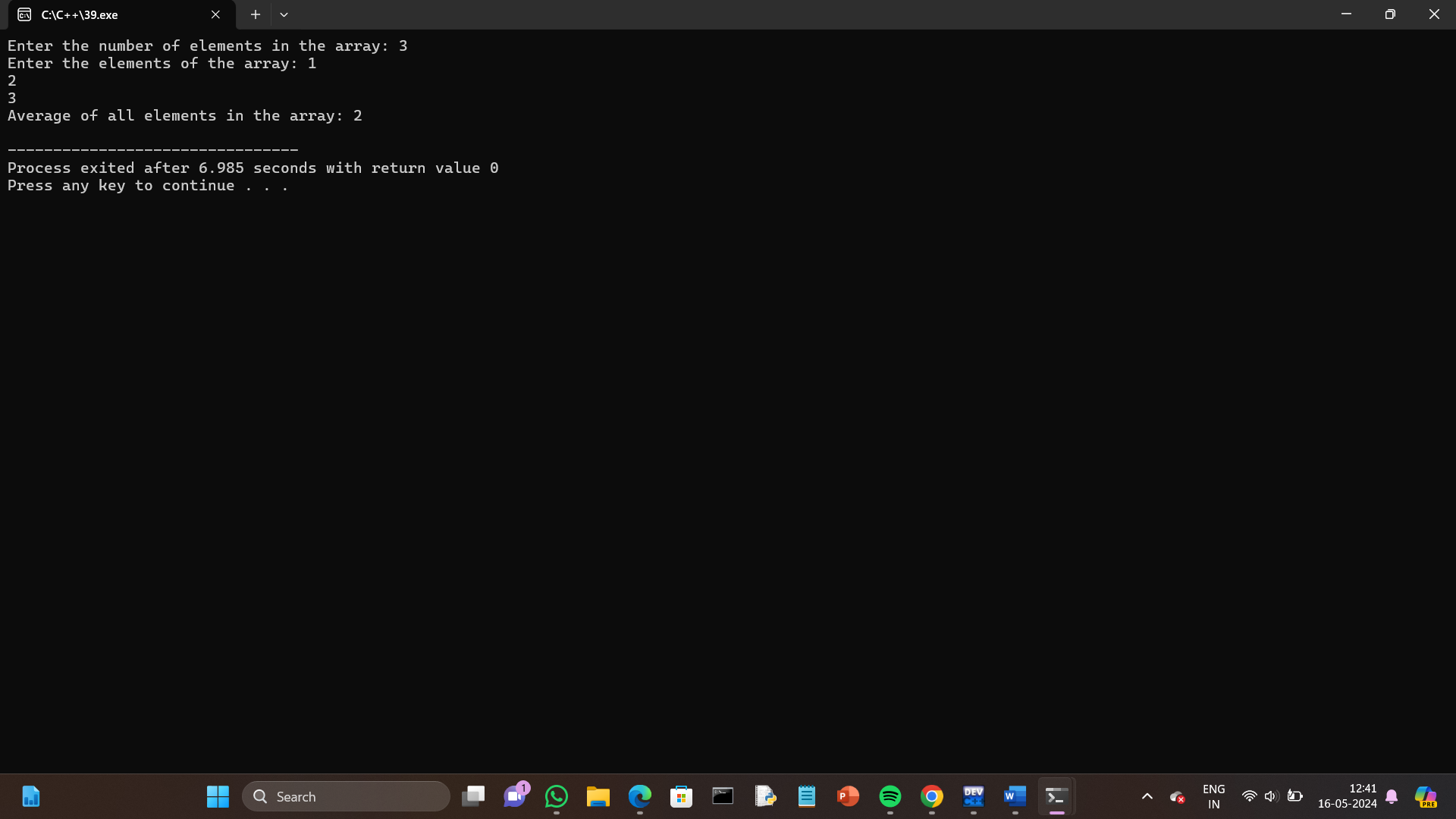
// Print the average of the elements

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

return 0;

}

Output:



40.Find the second largest element in an array.

Program:

#include <iostream>

#include <limits.h> // For INT\_MIN

using namespace std;

int main() {

int n;

// Take the number of elements in the array from the user

cout << "Enter the number of elements in the array: ";

cin >> n;

// Check if there are at least two elements

if (n < 2) {

cout << "Array should have at least two elements." << endl;

return 1;

}

int arr[n];

// Take array elements input from the user

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

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

cin >> arr[i];

}

// Initialize first and second largest to the minimum possible value

int firstLargest = INT\_MIN;

int secondLargest = INT\_MIN;

// Traverse the array to find the first and second largest elements

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

if (arr[i] > firstLargest) {

secondLargest = firstLargest;

firstLargest = arr[i];

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

secondLargest = arr[i];

}

}

if (secondLargest == INT\_MIN) {

cout << "There is no second largest element in the array." << endl;

} else {

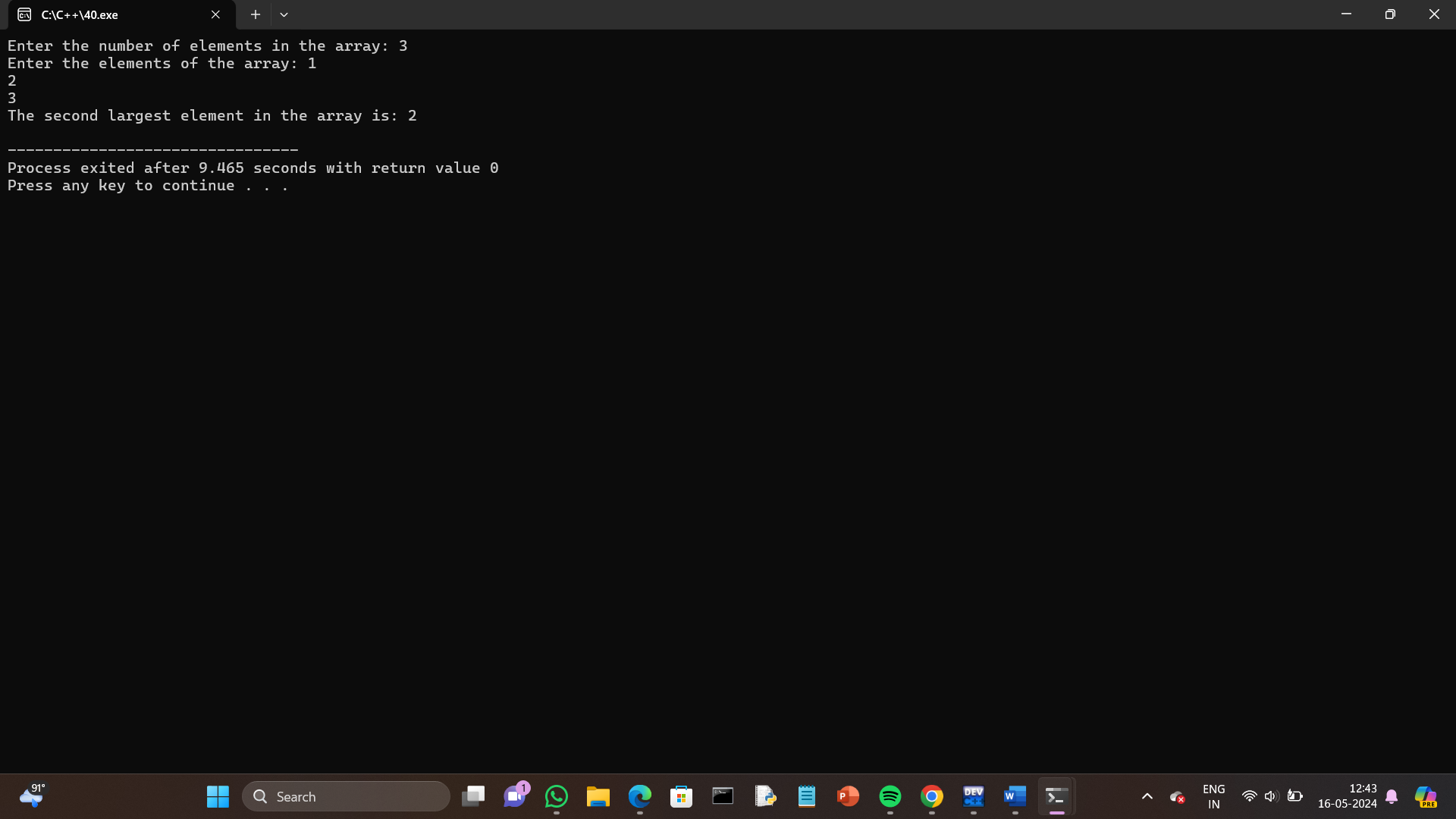
cout << "The second largest element in the array is: " << secondLargest << endl;

}

return 0;

}

Output:



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

Program:

#include <iostream>

using namespace std;

int main() {

int n, value, count = 0;

// Take the number of elements in the array from the user

cout << "Enter the number of elements in the array: ";

cin >> n;

int arr[n];

// Take array elements input from the user

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

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

cin >> arr[i];

}

// Take the value whose occurrences need to be counted

cout << "Enter the value to count its occurrences: ";

cin >> value;

// Count the occurrences of the value in the array

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

if (arr[i] == value) {

count++;

}

}

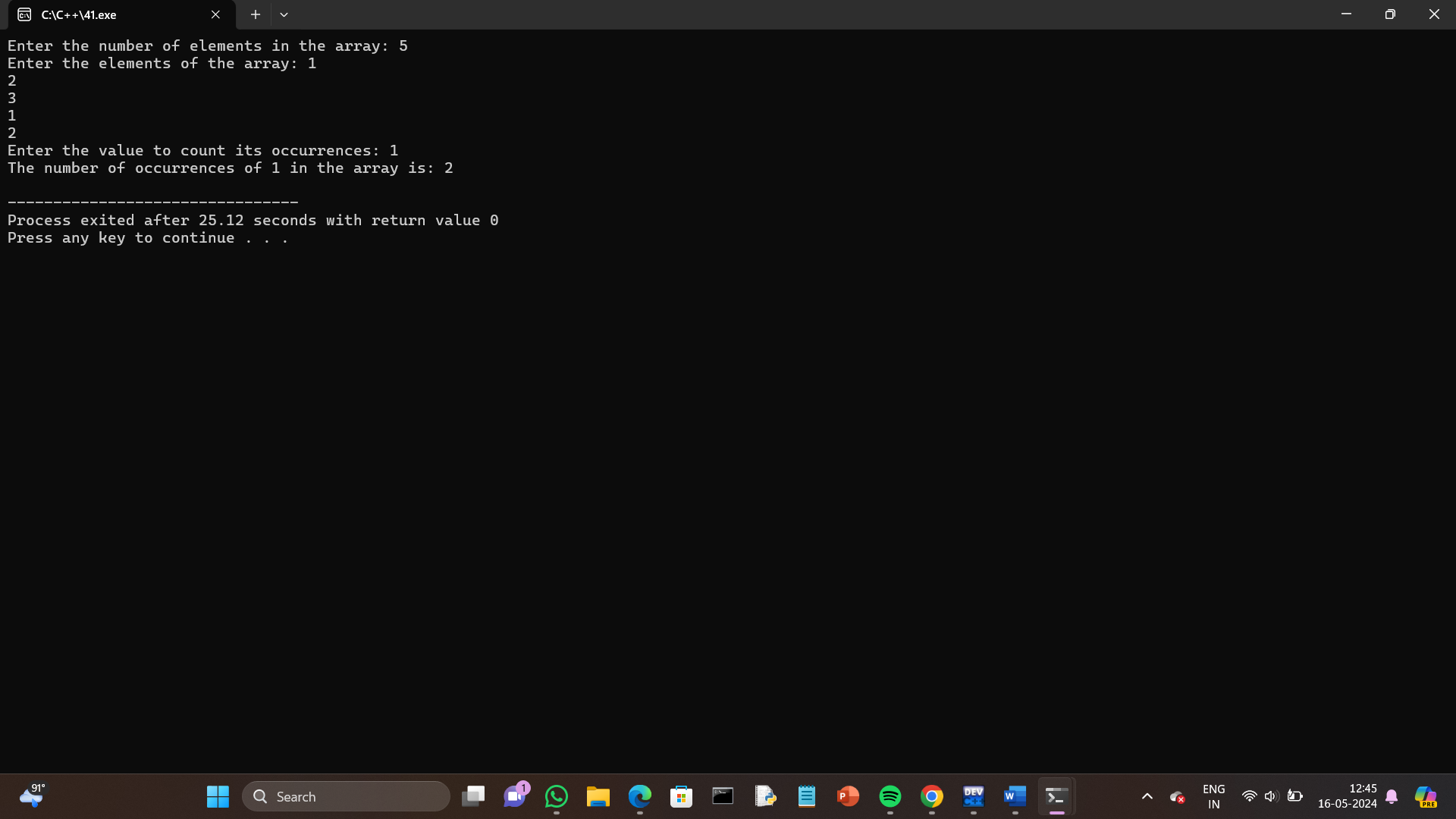
// Print the number of occurrences

cout << "The number of occurrences of " << value << " in the array is: " << count << endl;

return 0;

}

Output:



42.Merge two array.

Program:

#include <iostream>

using namespace std;

int main() {

int n1, n2;

// Take the number of elements in the first array from the user

cout << "Enter the number of elements in the first array: ";

cin >> n1;

int arr1[n1];

// Take elements of the first array from the user

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

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

cin >> arr1[i];

}

// Take the number of elements in the second array from the user

cout << "Enter the number of elements in the second array: ";

cin >> n2;

int arr2[n2];

// Take elements of the second array from the user

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

for (int i = 0; i < n2; i++) {

cin >> arr2[i];

}

// Create a merged array of size n1 + n2

int merged[n1 + n2];

// Copy elements from the first array to the merged array

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

merged[i] = arr1[i];

}

// Copy elements from the second array to the merged array

for (int i = 0; i < n2; i++) {

merged[n1 + i] = arr2[i];

}

// Print the merged array

cout << "Merged array: ";

for (int i = 0; i < n1 + n2; i++) {

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

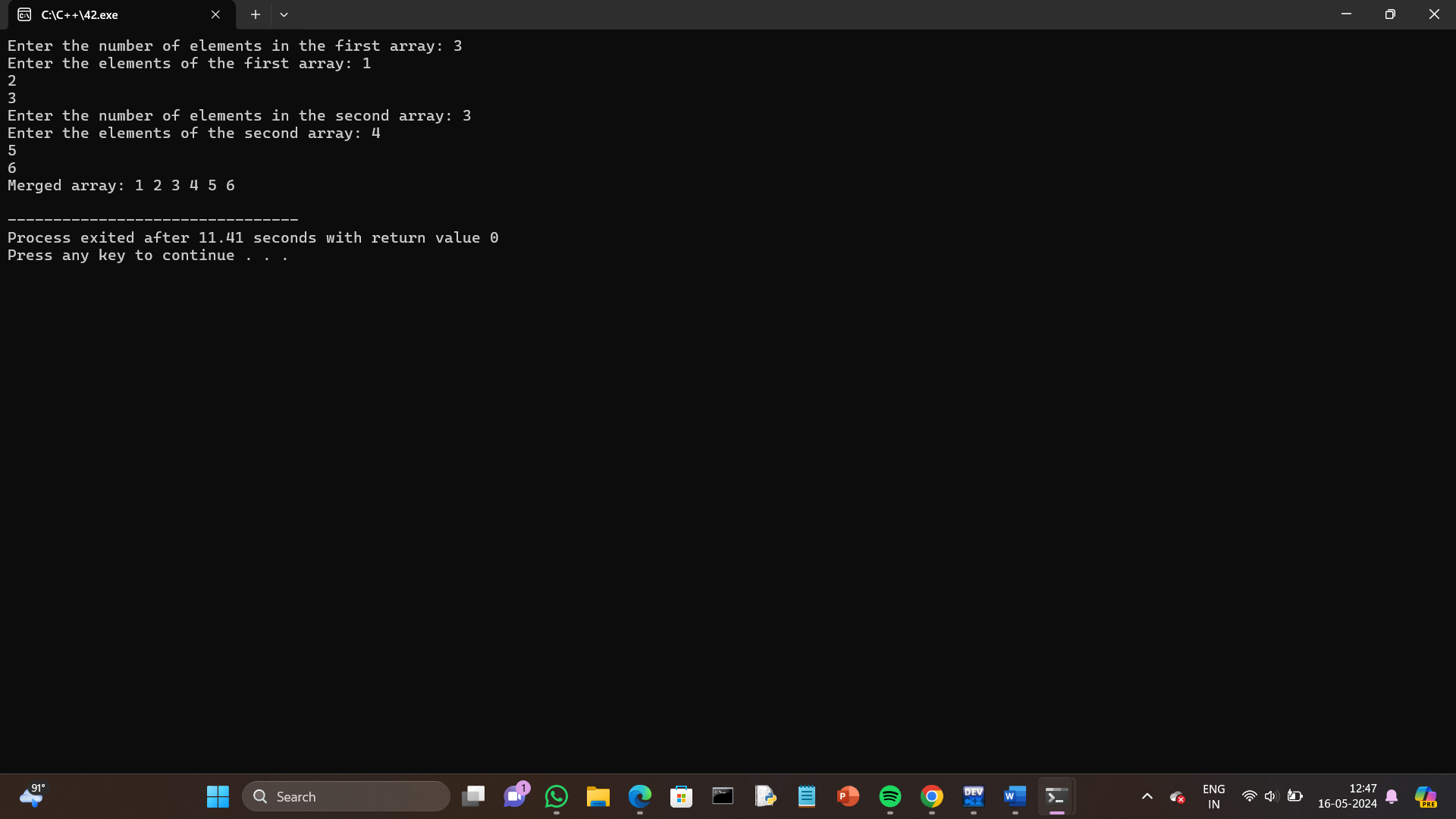
}

cout << endl;

return 0;

}

Output:



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

Program:

#include <iostream>

using namespace std;

int main() {

int n;

// Take the number of elements in the array from the user

cout << "Enter the number of elements in the array: ";

cin >> n;

// Dynamically allocate memory for the array

int \*arr = new int[n];

// Take array elements input from the user

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

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

cin >> arr[i];

}

// Display the values of the dynamic array

cout << "Values of the dynamic array: ";

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

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

}

cout << endl;

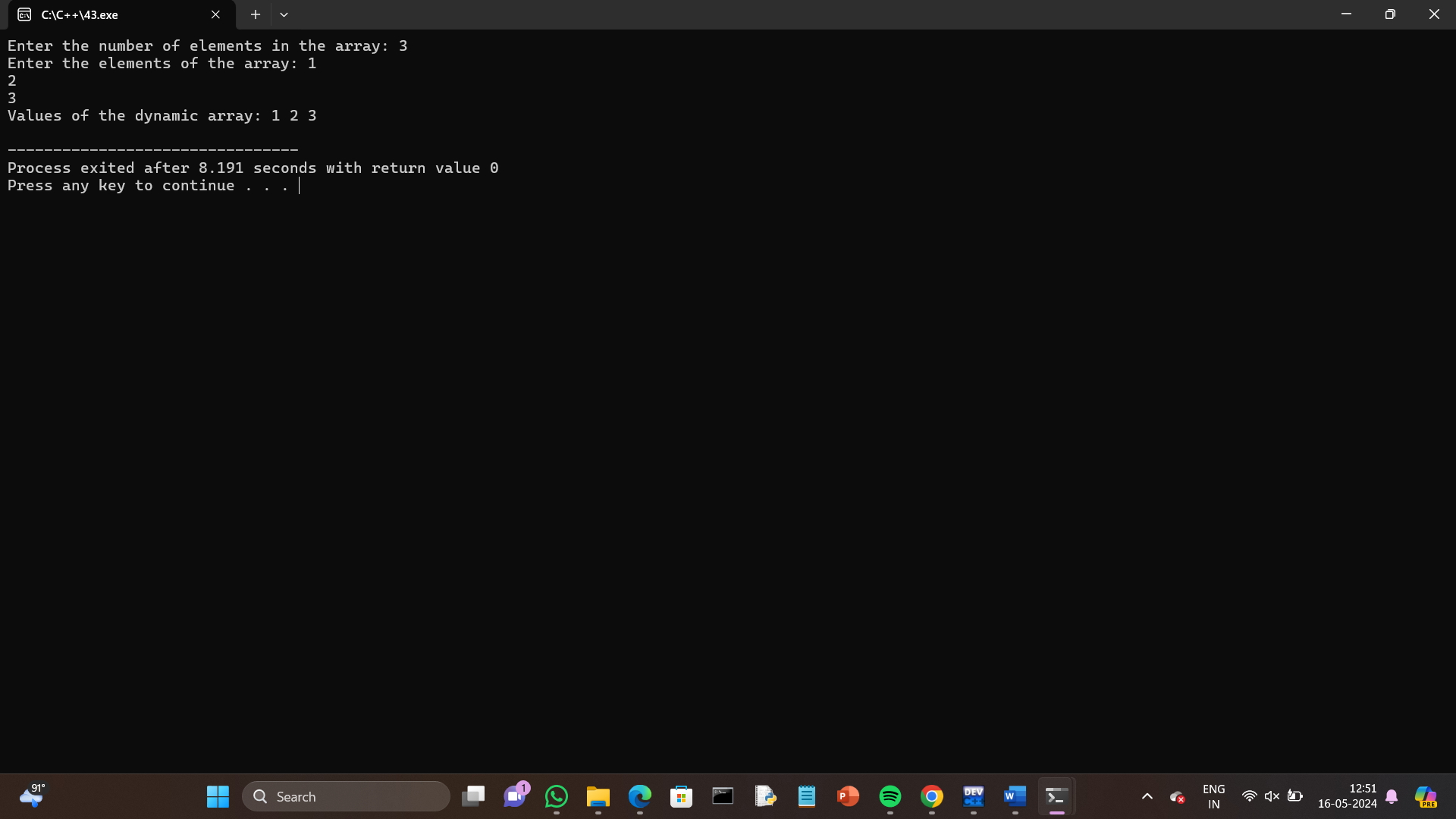
// Deallocate memory for the dynamic array

delete[] arr;

return 0;

}

Output:



44.Add 2 matrices.

Program:

#include <iostream>

using namespace std;

int main() {

int rows, cols;

// Take input for number of rows and columns of matrices

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

cin >> rows >> cols;

// Define two matrices with the specified size

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

// Input elements of the first matrix

cout << "Enter the elements of the first matrix:" << endl;

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

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

cin >> matrix1[i][j];

}

}

// Input elements of the second matrix

cout << "Enter the elements of the second matrix:" << endl;

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

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

cin >> matrix2[i][j];

}

}

// Perform matrix addition

cout << "Result of matrix addition:" << endl;

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[i][j] << " ";

}

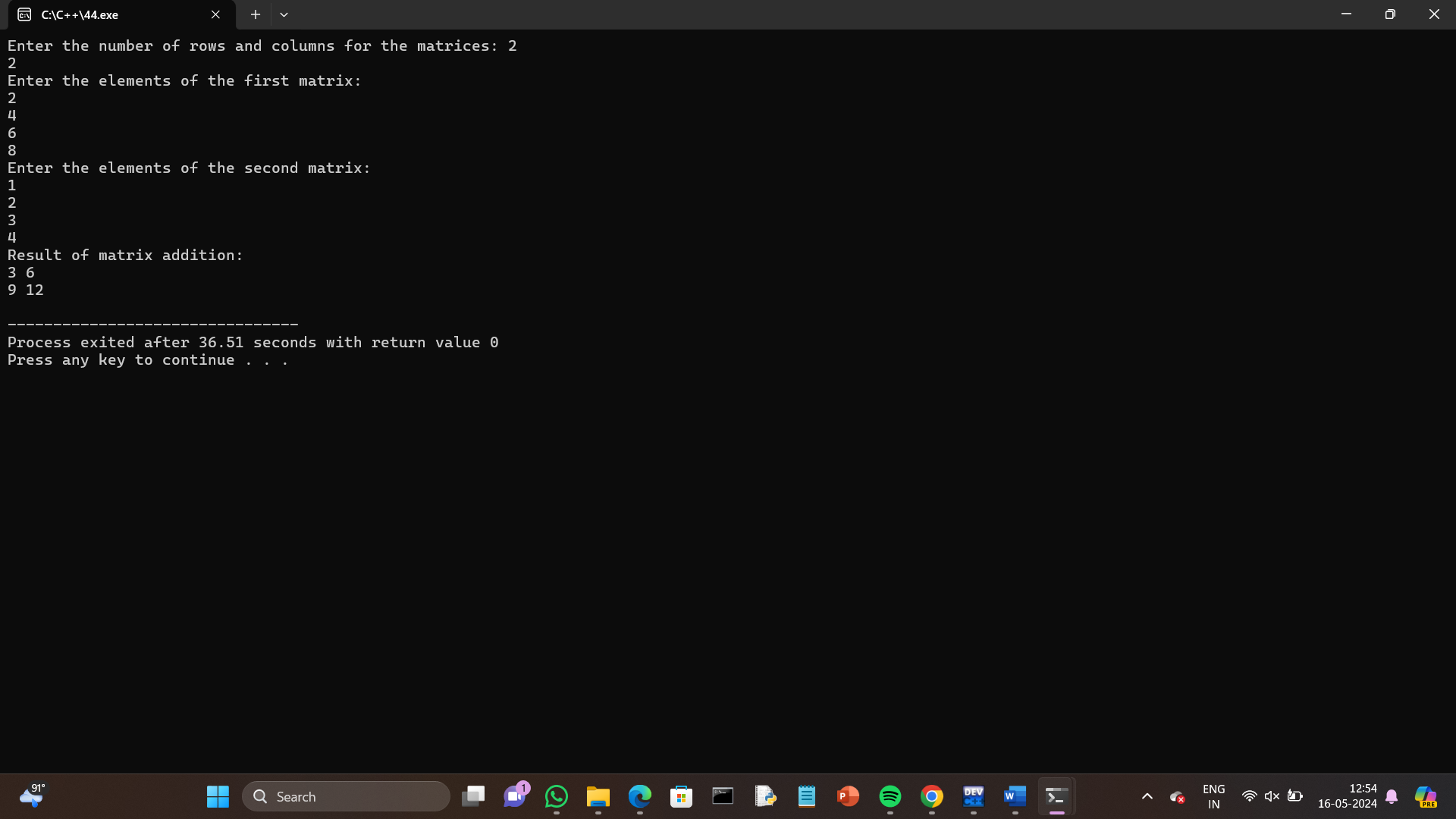
cout << endl;

}

return 0;

}

Output:



45.Multiply 2 matrices.

Program:

#include <iostream>

using namespace std;

int main() {

int rows1, cols1, rows2, cols2;

// Input for dimensions of the first matrix

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

cin >> rows1 >> cols1;

// Input for dimensions of the second matrix

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

cin >> rows2 >> cols2;

// Check if multiplication is possible

if (cols1 != rows2) {

cout << "Matrix multiplication is not possible. Number of columns in the first matrix should be equal to the number of rows in the second matrix." << endl;

return 1;

}

// Define two matrices with the specified sizes

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

// Input elements of the first matrix

cout << "Enter the elements of the first matrix:" << endl;

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

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

cin >> matrix1[i][j];

}

}

// Input elements of the second matrix

cout << "Enter the elements of the second matrix:" << endl;

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

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

cin >> matrix2[i][j];

}

}

// Perform matrix multiplication

cout << "Result of matrix multiplication:" << endl;

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

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

product[i][j] = 0;

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

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

}

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

}

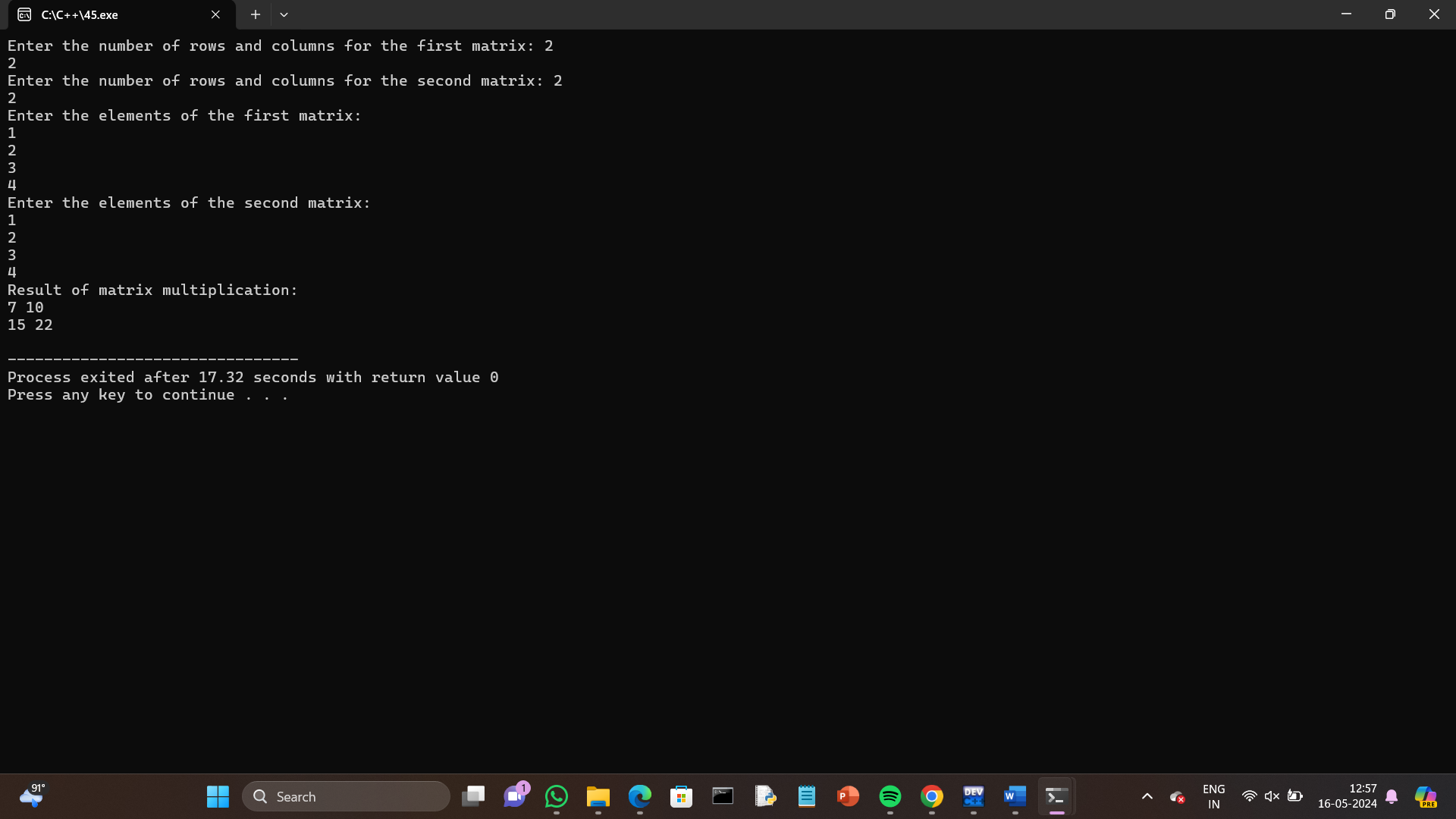
cout << endl;

}

return 0;

}

Output:



46.Find the sum of diagonals of a matrix.

Program:

#include <iostream>

using namespace std;

int main() {

int rows, cols;

// Input for dimensions of the matrix

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

cin >> rows >> cols;

// Check if it's a square matrix

if (rows != cols) {

cout << "Matrix should be square for diagonal sum calculation." << endl;

return 1;

}

// Define a matrix with the specified size

int matrix[rows][cols];

// Input elements of the matrix

cout << "Enter the elements of the matrix:" << endl;

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

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

cin >> matrix[i][j];

}

}

// Calculate the sum of the main diagonal

int mainDiagonalSum = 0;

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

mainDiagonalSum += matrix[i][i];

}

// Calculate the sum of the secondary diagonal

int secondaryDiagonalSum = 0;

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

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

}

// Print the sum of diagonals

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

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

return 0;

}

Output:

