1. Write a C program to print an array.

Code -

#include <stdio.h>

void printArray(int arr[], int size) {

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

printf("%d ", arr[i]);

}

printf("\n");

}

int main() {

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

int size = sizeof(myArray) / sizeof(myArray[0]);

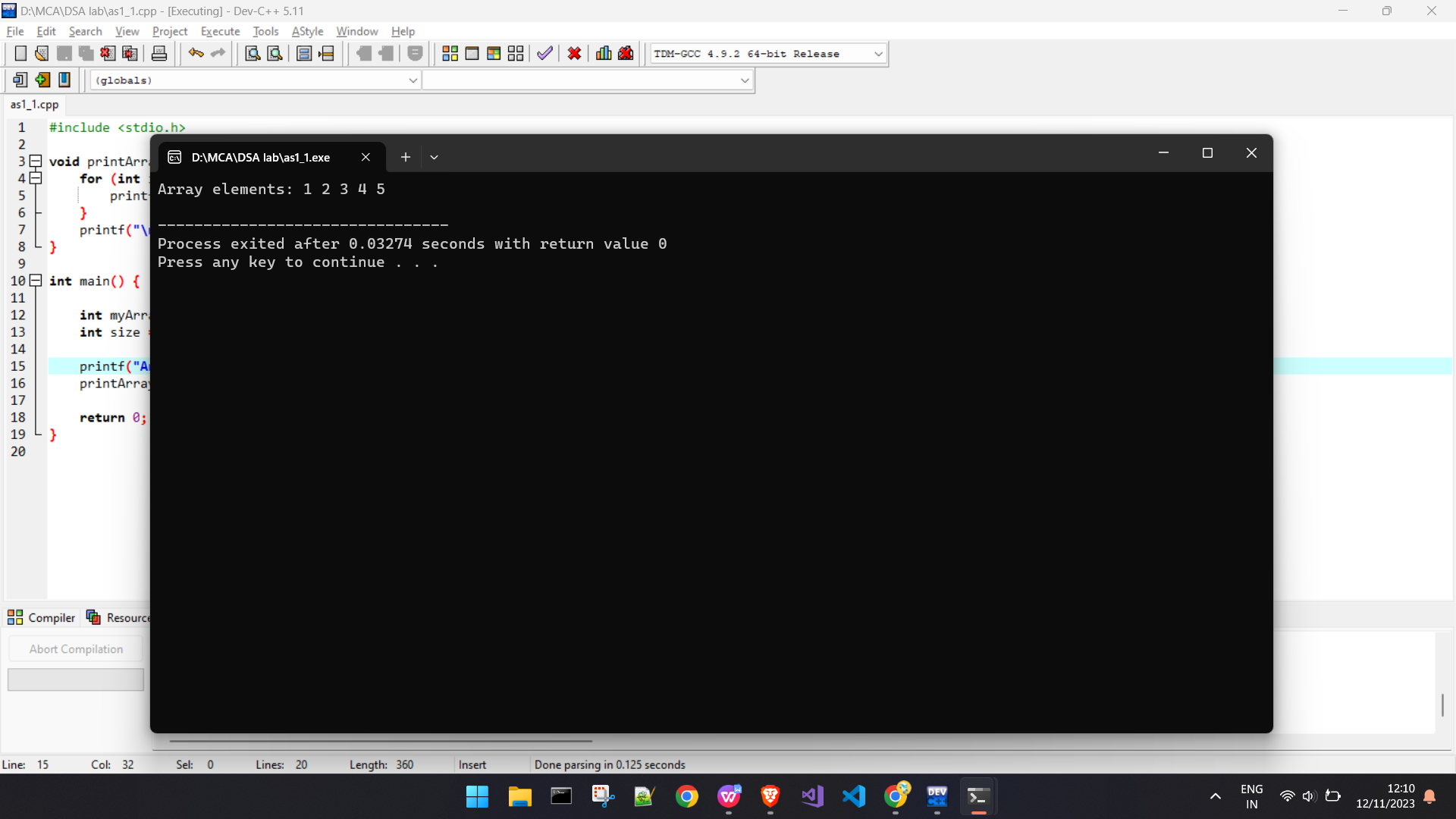
printf("Array elements: ");

printArray(myArray, size);

return 0;

}

Output -



1. Write a C program to check whether a given string is Palindrome or not.

Code -

#include <stdio.h>

#include <string.h>

int main()

{

char str[10] = "mam";

int i, len, flag = 0;

len = strlen(str);

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

{

if (str[i] != str[len - i - 1]) {

flag = 1;

break;

}

}

if (flag)

printf("%s is not palindrome", str);

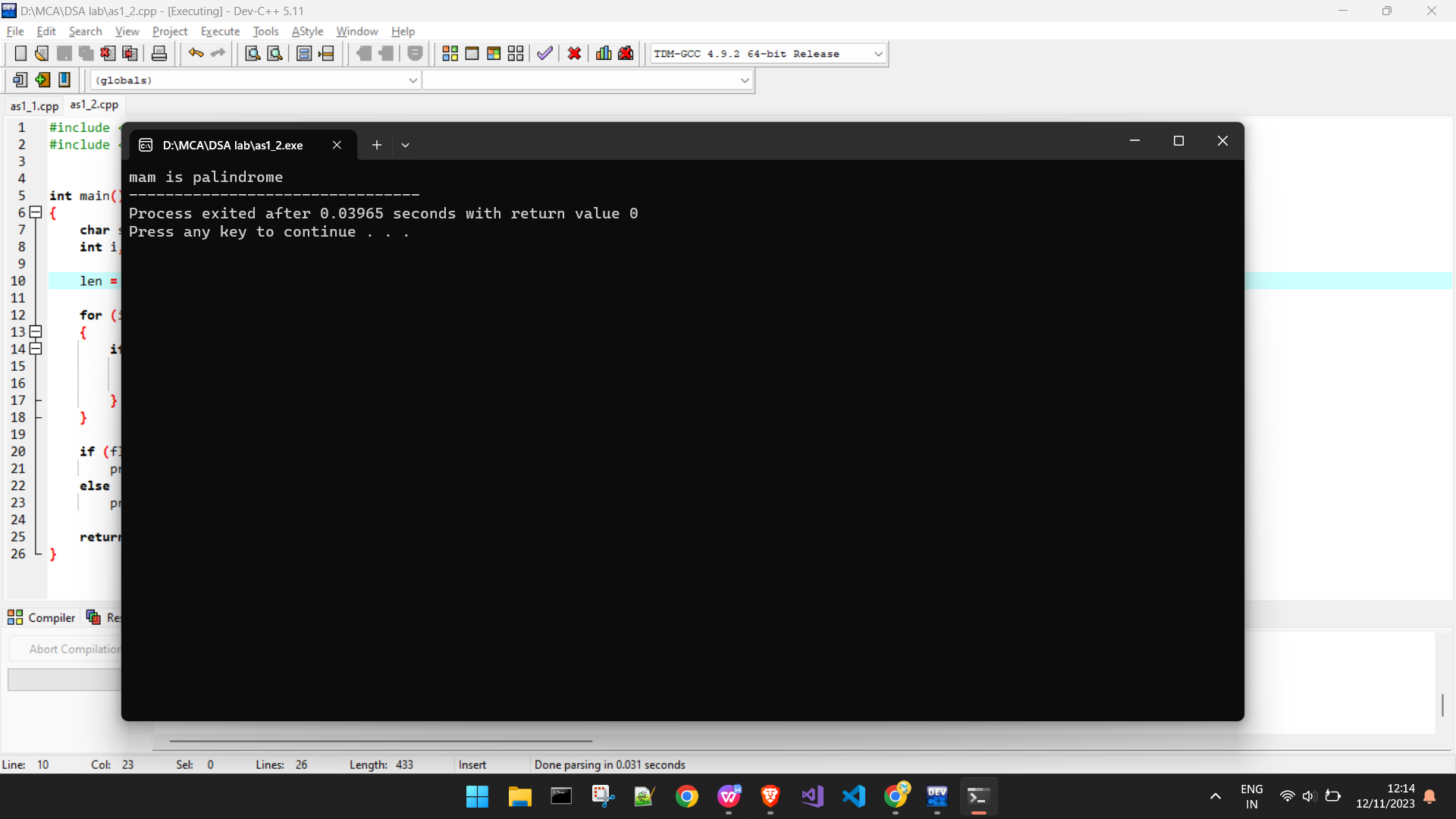
else

printf("%s is palindrome", str);

return 0;

}

Output -



1. Write a C program to convert temperature from degree Centigrade to Fahrenheit.

Code -

#include<stdio.h>

int main()

{

float celsius,fahrenheit;

printf("\n Enter the Temparature in Celsius : ");

scanf("%f",&celsius);

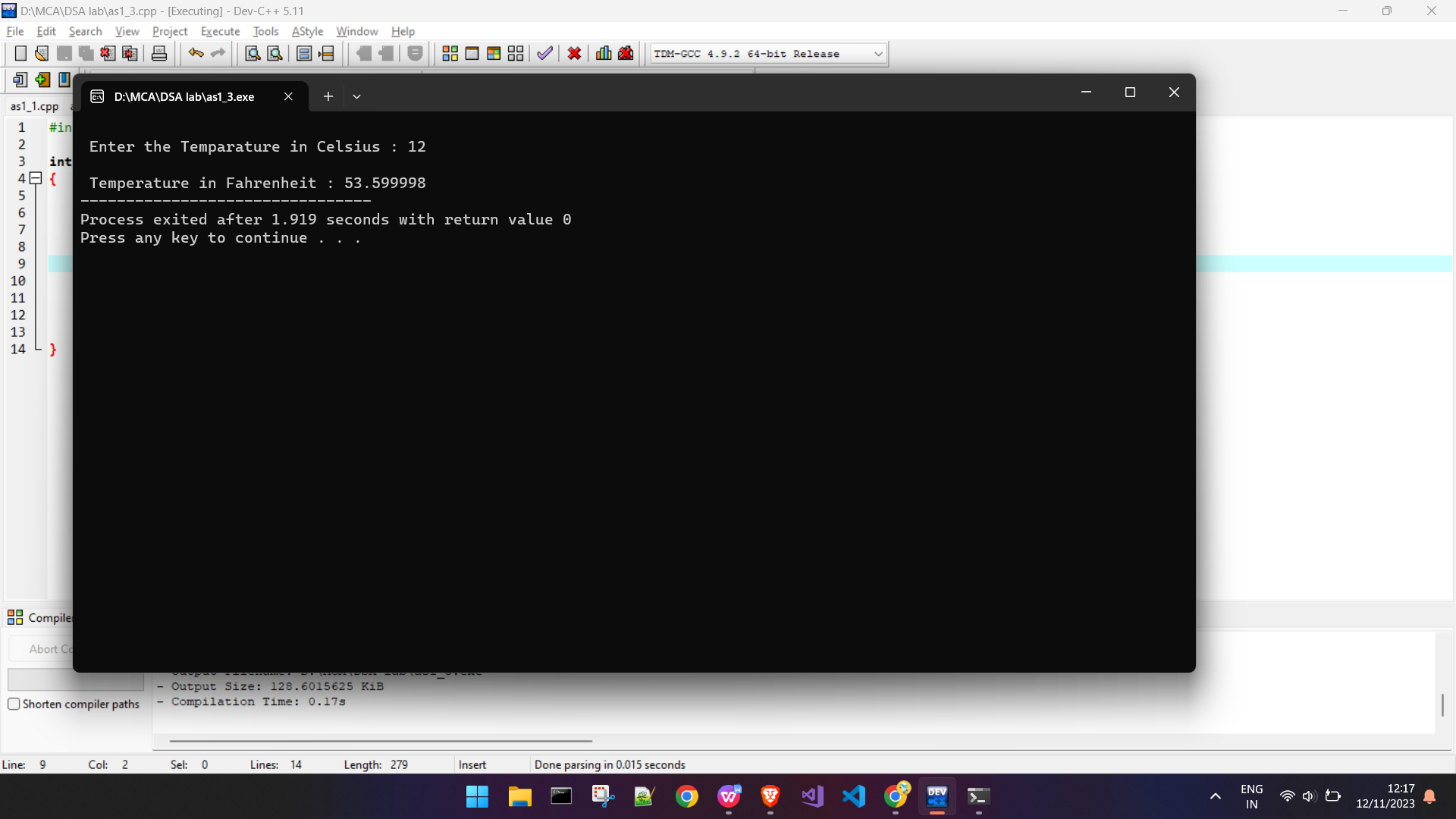
fahrenheit = (1.8 \* celsius) + 32;

printf("\n Temperature in Fahrenheit : %f ",fahrenheit);

return 0 ;

}

Output -



1. Write a C program to sort an array.

Code -

#include <stdio.h>

int main()

{

int arr[] = {50, 20, 80, 70, 10};

int temp = 0;

int length = sizeof(arr)/sizeof(arr[0]);

printf("Elements of original array: \n");

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

printf("%d ", arr[i]);

}

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

for (int j = i+1; j < length; j++) {

if(arr[i] > arr[j]) {

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

printf("\n");

printf("Elements of array sorted in ascending order: \n");

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

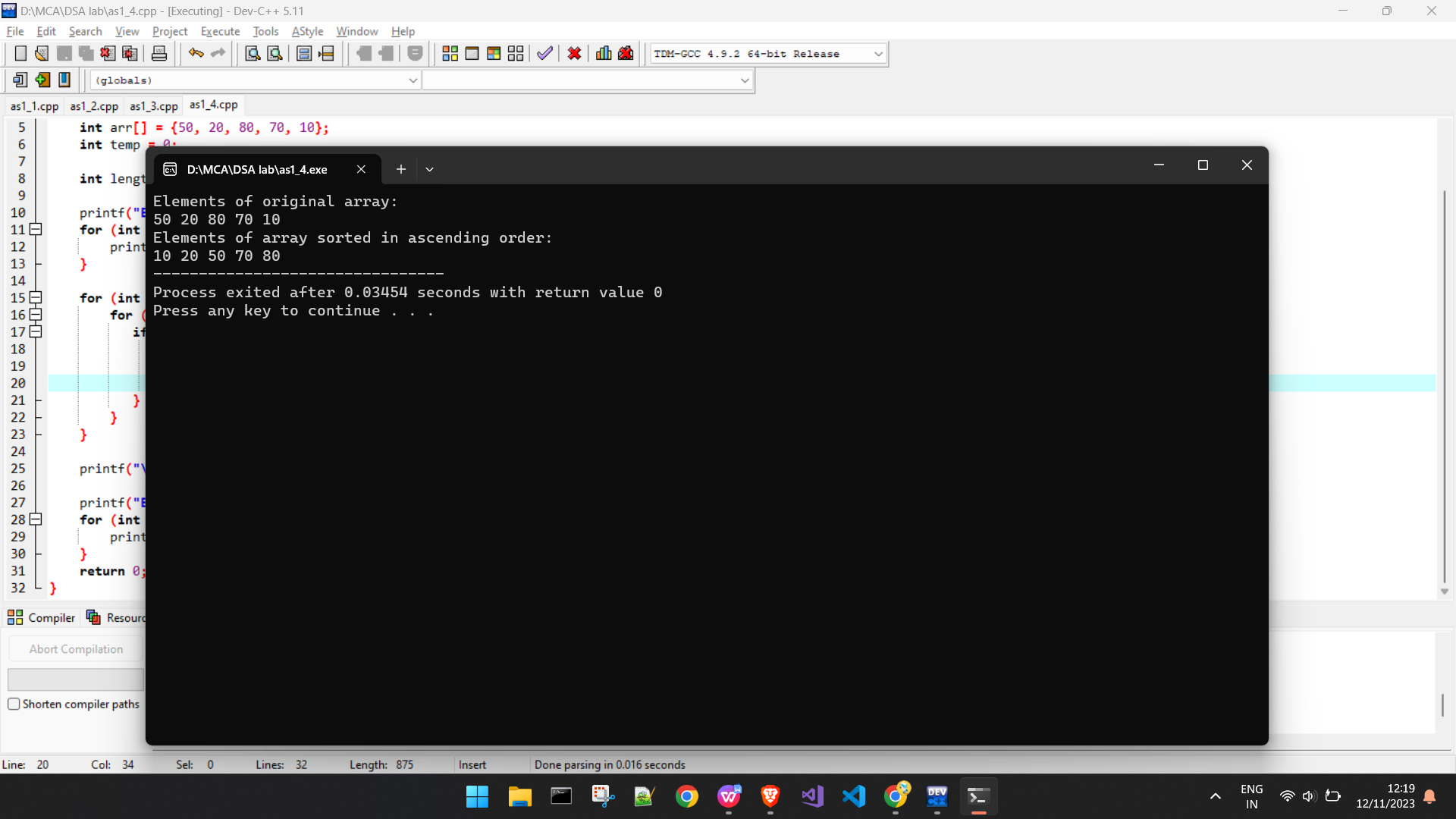
printf("%d ", arr[i]);

}

return 0;

}

Output -



1. Write a C program to print the largest and second largest element of the array.

Code -

#include<stdio.h>

int main()

{

int arr[] = {50, 20, 80, 70, 10};

int temp = 0;

int length = sizeof(arr)/sizeof(arr[0]);

printf("Elements of original array: \n");

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

printf("%d ", arr[i]);

}

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

for (int j = i+1; j < length; j++) {

if(arr[i] > arr[j]) {

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

int secondL = arr[length-2];

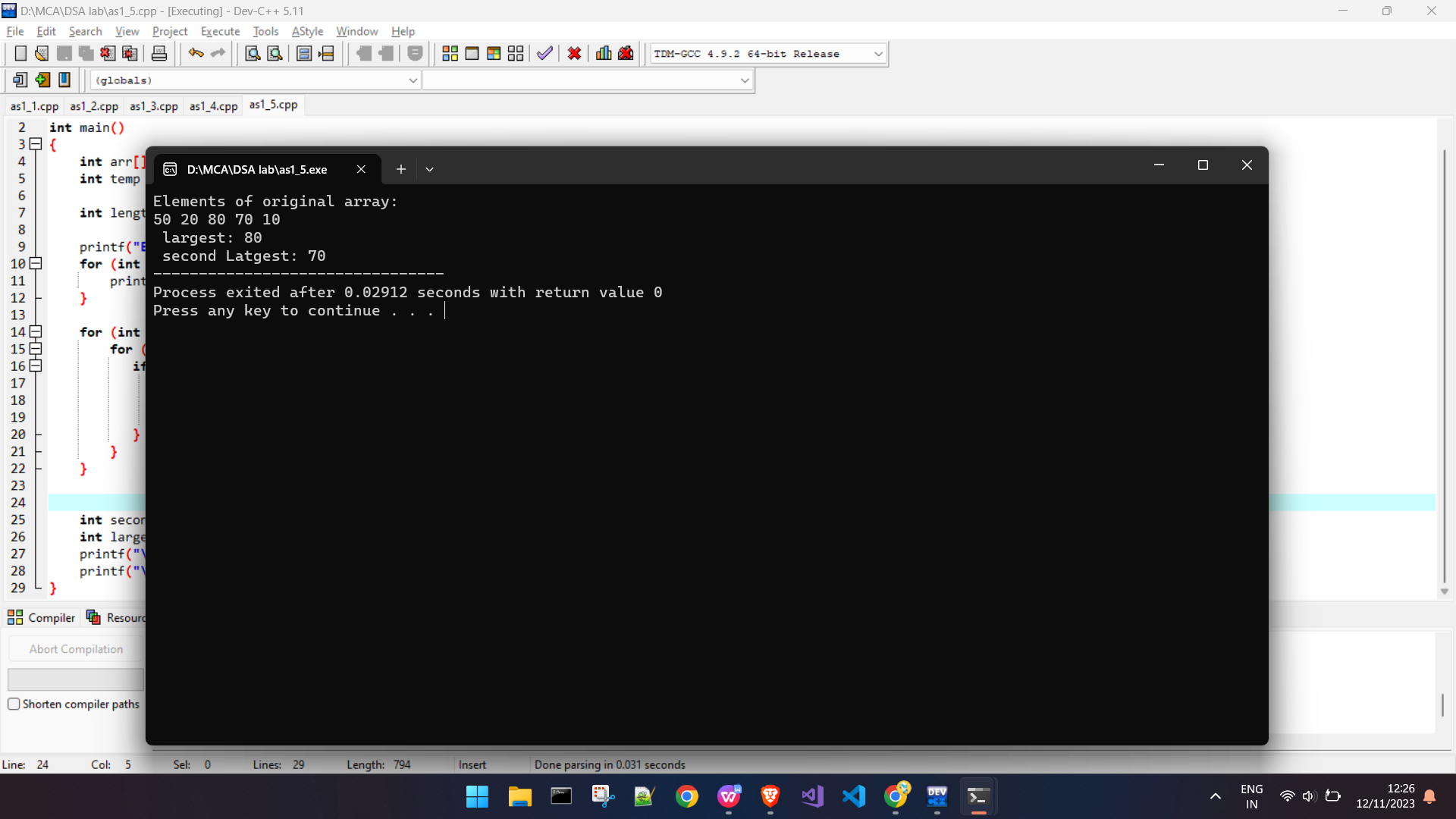
int largest = arr[length-1];

printf("\n largest: %d",largest);

printf("\n second Latgest: %d",secondL);

}

Output -



1. Write a C program to display Fibonacci series.

Code -

#include<stdio.h>

int main()

{

int n1=0,n2=1,n3,i,number;

printf("Enter the number of elements:");

scanf("%d",&number);

printf("\n%d %d",n1,n2);

for(i=2;i<number;++i)

{

n3=n1+n2;

printf(" %d",n3);

n1=n2;

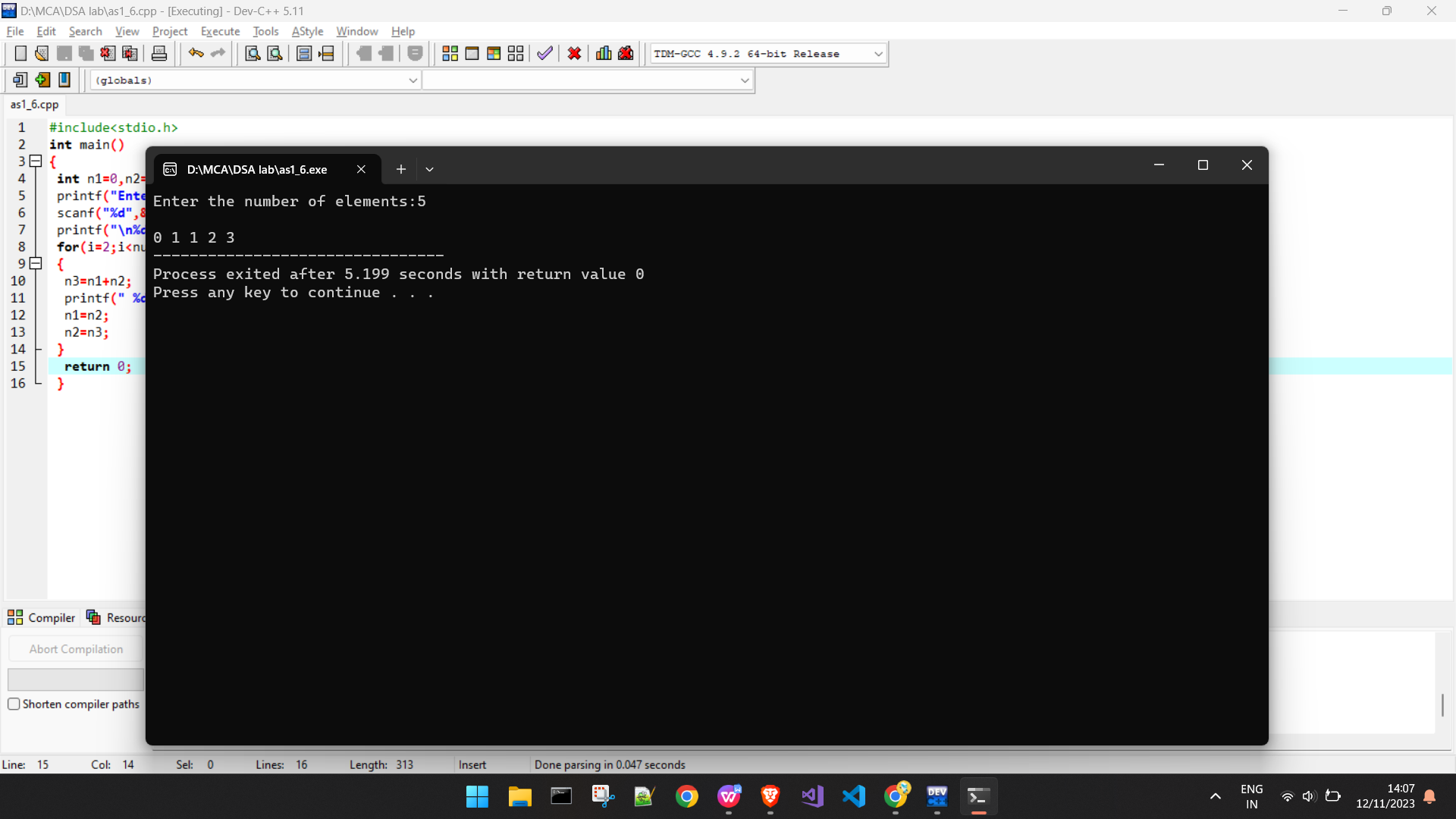
n2=n3;

}

return 0;

}

Output -



1. Write a program that reads two 2D metrices from the console, verifies if metrics multiplication is possible or not. Then multiplies the metrices and prints the 3rd metrics.

Code -

#include <stdio.h>

void readMatrix(int matrix[10][10], int rows, int cols) {

printf("Enter the elements of the matrix:\n");

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

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

scanf("%d", &matrix[i][j]);

}

}

}

void displayMatrix(int matrix[10][10], int rows, int cols) {

printf("Matrix:\n");

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

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

printf("%d\t", matrix[i][j]);

}

printf("\n");

}

}

int isMultiplicationPossible(int cols1, int rows2) {

return cols1 == rows2;

}

void multiplyMatrices(int mat1[10][10], int mat2[10][10], int result[10][10], int rows1, int cols1, int rows2, int cols2) {

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

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

result[i][j] = 0;

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

result[i][j] += mat1[i][k] \* mat2[k][j];

}

}

}

}

int main() {

int mat1[10][10], mat2[10][10], result[10][10];

int rows1, cols1, rows2, cols2;

printf("Enter the number of rows for the first matrix: ");

scanf("%d", &rows1);

printf("Enter the number of columns for the first matrix: ");

scanf("%d", &cols1);

printf("Enter the number of rows for the second matrix: ");

scanf("%d", &rows2);

printf("Enter the number of columns for the second matrix: ");

scanf("%d", &cols2);

if (!isMultiplicationPossible(cols1, rows2)) {

printf("Matrix multiplication is not possible!\n");

return 1;

}

readMatrix(mat1, rows1, cols1);

readMatrix(mat2, rows2, cols2);

multiplyMatrices(mat1, mat2, result, rows1, cols1, rows2, cols2);

printf("\nMatrix 1:\n");

displayMatrix(mat1, rows1, cols1);

printf("\nMatrix 2:\n");

displayMatrix(mat2, rows2, cols2);

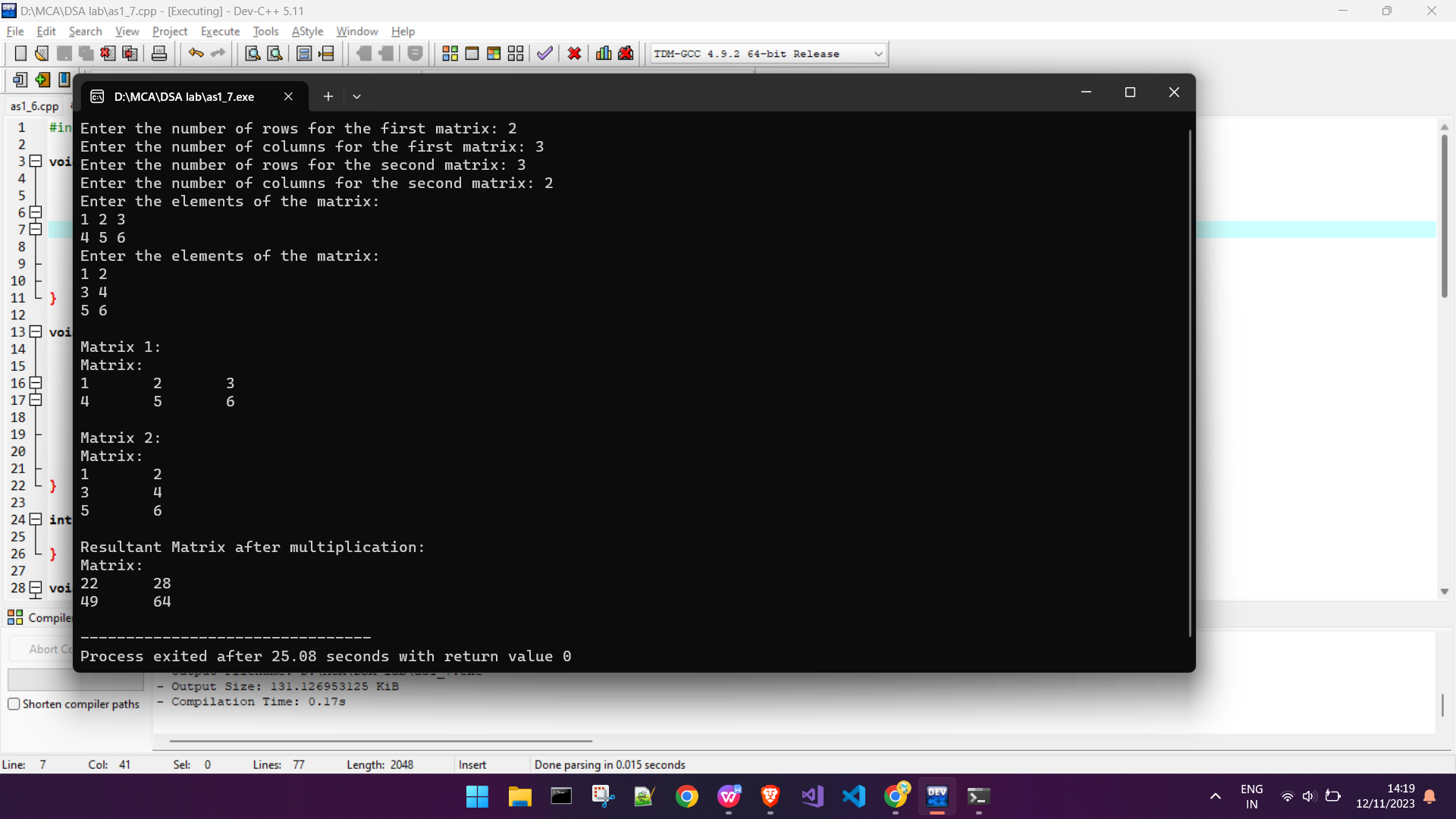
printf("\nResultant Matrix after multiplication:\n");

displayMatrix(result, rows1, cols2);

return 0;

}

Output -



1. Write a program that reads a 2D metrics and checks if the metrics is a symmetric metrics or not.

Code -

#include <stdio.h>

int main() {

int a[3][3] = {{1, 2, 3}, {2, 4, 5}, {3, 5, 6}};

int i, j, flag = 0;

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

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

if (a[i][j] != a[j][i]) {

flag = 1;

break;

}

}

if (flag == 1) {

break;

}

}

if (flag == 0) {

printf("Matrix is symmetric\n");

} else {

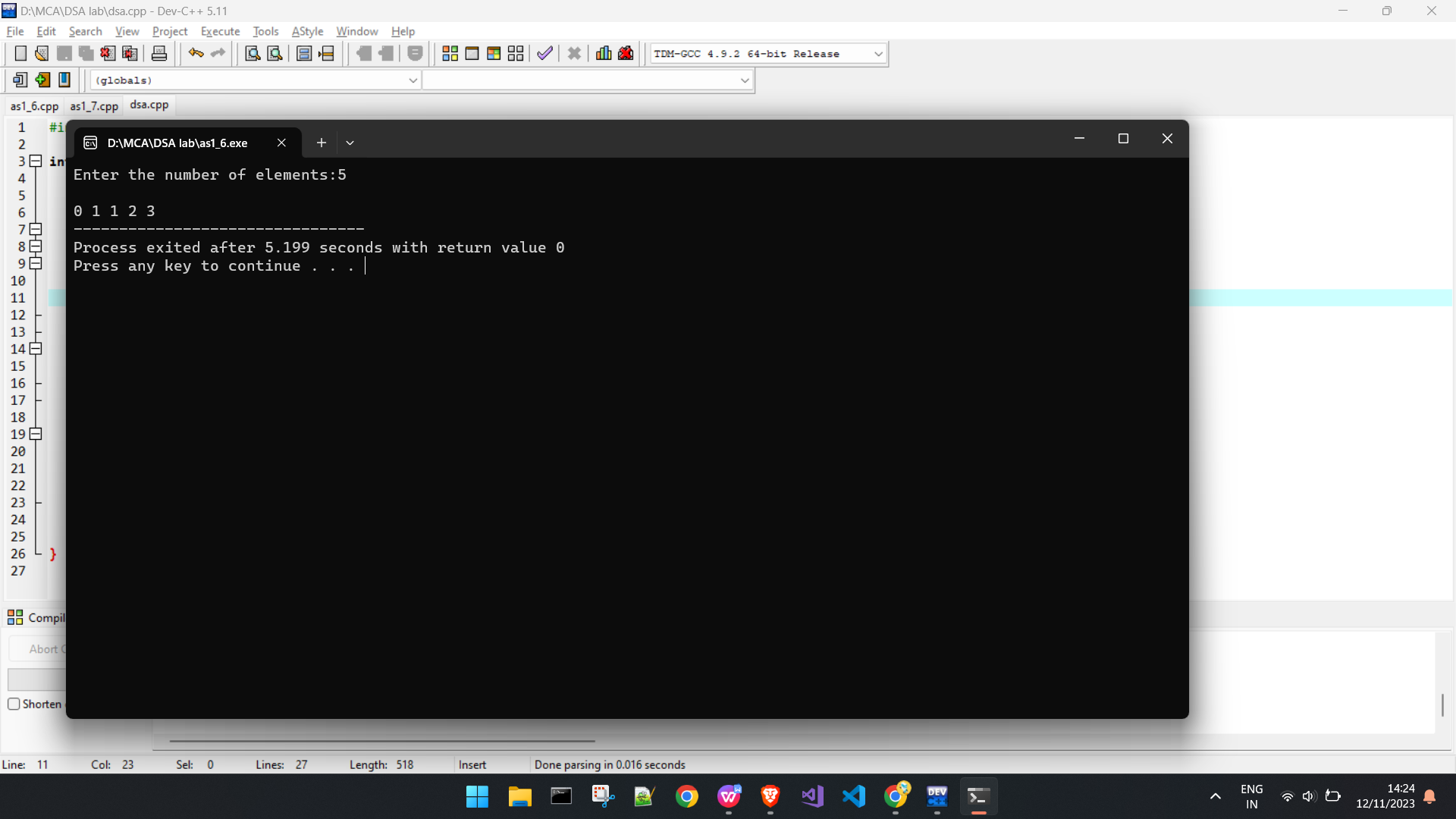
printf("Matrix is not symmetric\n");

}

return 0;

}

Output -



1. Write a C program to print reverse array

Code -

#include <stdio.h>

int main() {

int size = 5,i;

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

printf("Original Array: ");

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

printf("%d ", arr[i]);

}

printf("\n");

printf("\nReverse Array: ");

for(i = size-1; i>= 0; i--){

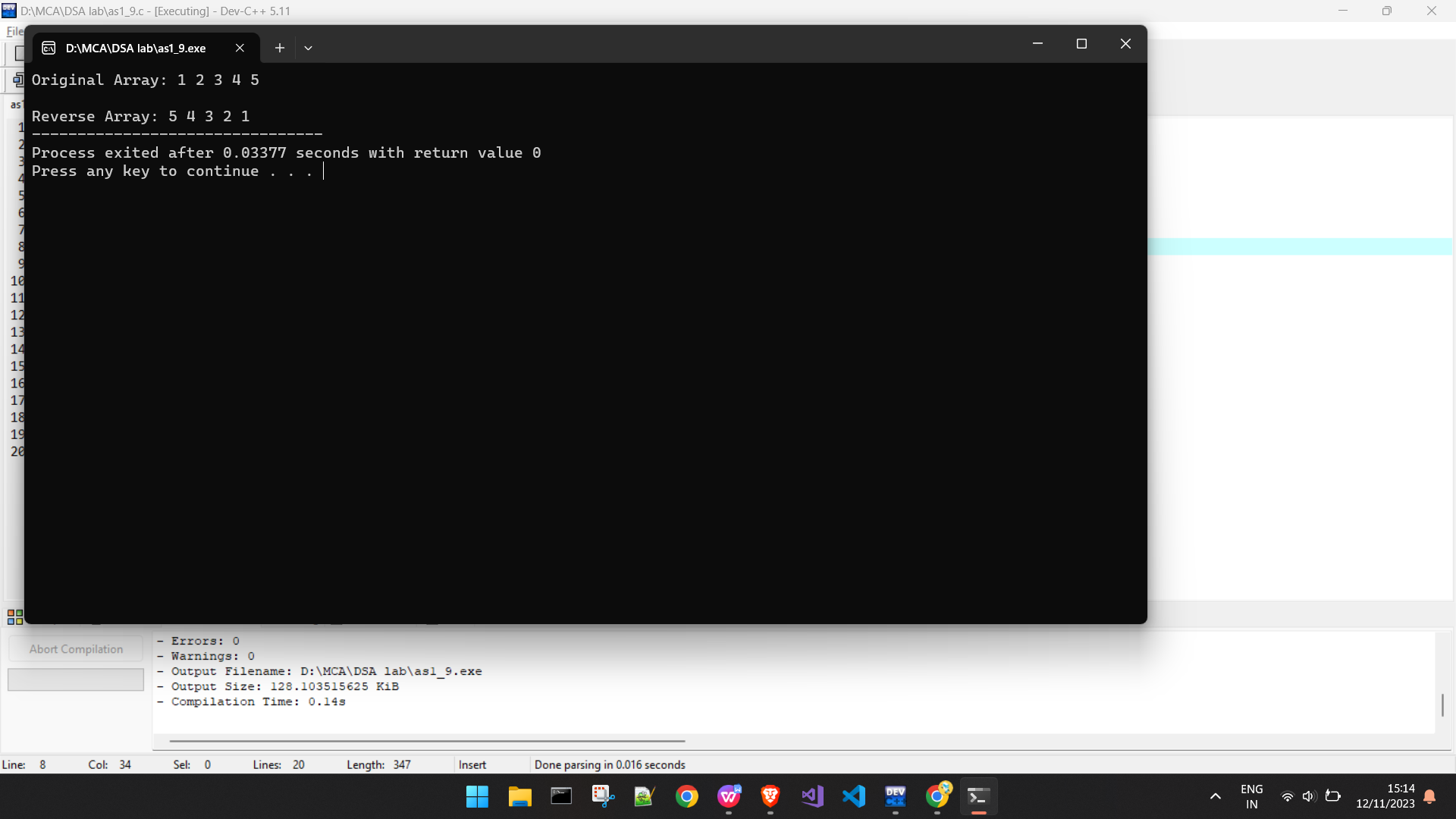
printf("%d ",arr[i]);

}

return 0;

}

Output -



1. Write a C program to check the sum of all elements of an array

Code -

#include <stdio.h>

int main() {

int size = 5;

int sum = 0;

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

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

sum += arr[i];

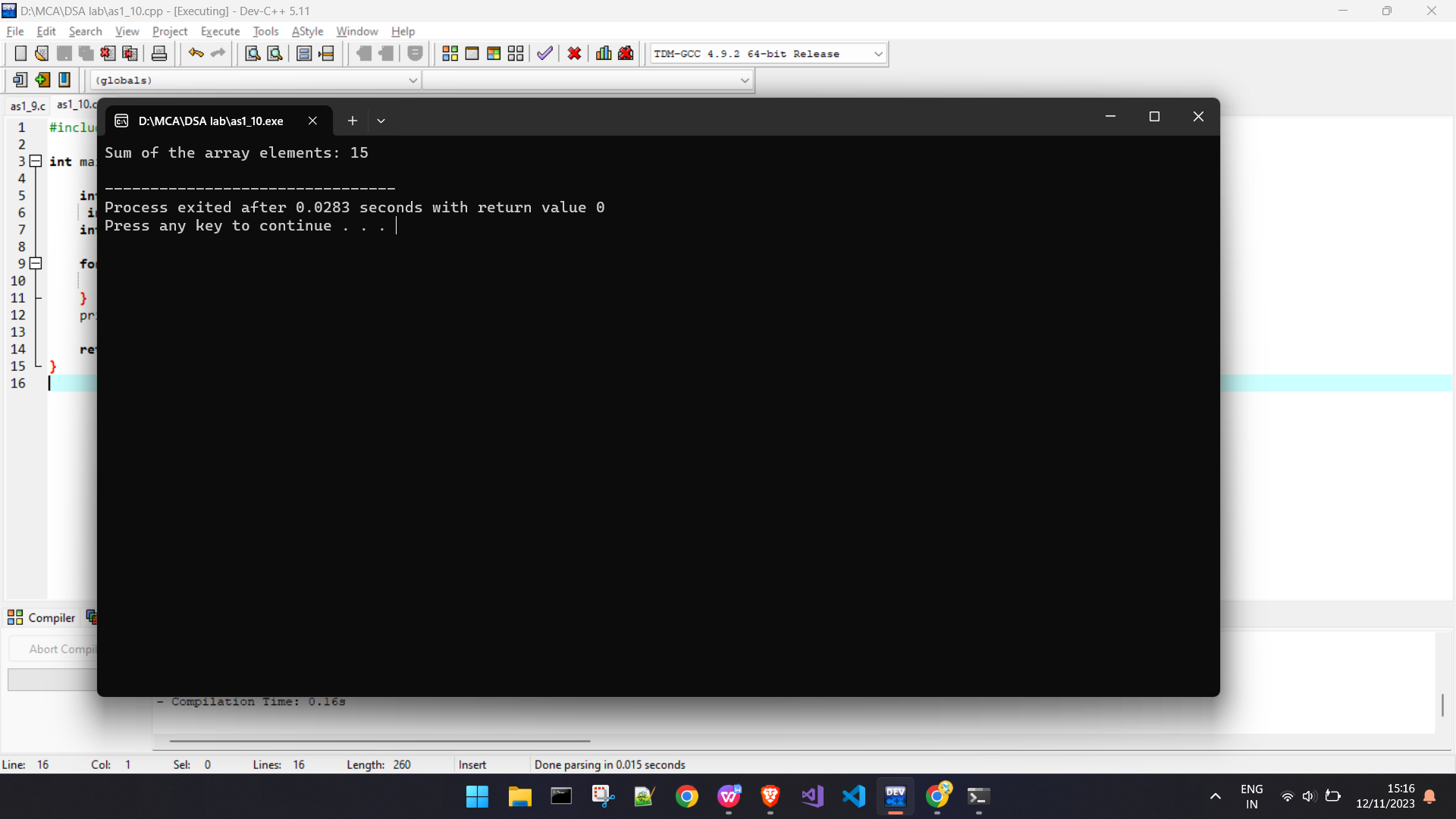
}

printf("Sum of the array elements: %d\n", sum);

return 0;

}

Output -



11.11. Write a C program to check duplicate number in an array.

Code-

#include <stdio.h>

int main() {

int size = 5,flag=0;

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

for (int i = 0; i < size - 1; i++) {

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

if (arr[i] == arr[j]) {

flag = 1;

}

}

}

if(flag){

printf("Array has duplicate elements");

}else{

printf("Array has no duplicate elements");

}

return 0;

}

Output -

