**DAY 2 EXPERIMENTS**

1)COMPOSITE NUMBER:

#include<stdio.h>

int main (){

int i,n,a[100],count=0;

printf("enter size:");

scanf("%d",&n);

printf("enter elements\n");

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

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

}

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

if(a[i]==2){

continue;

}

else if(a[i]%2==0){

count++;

}

}

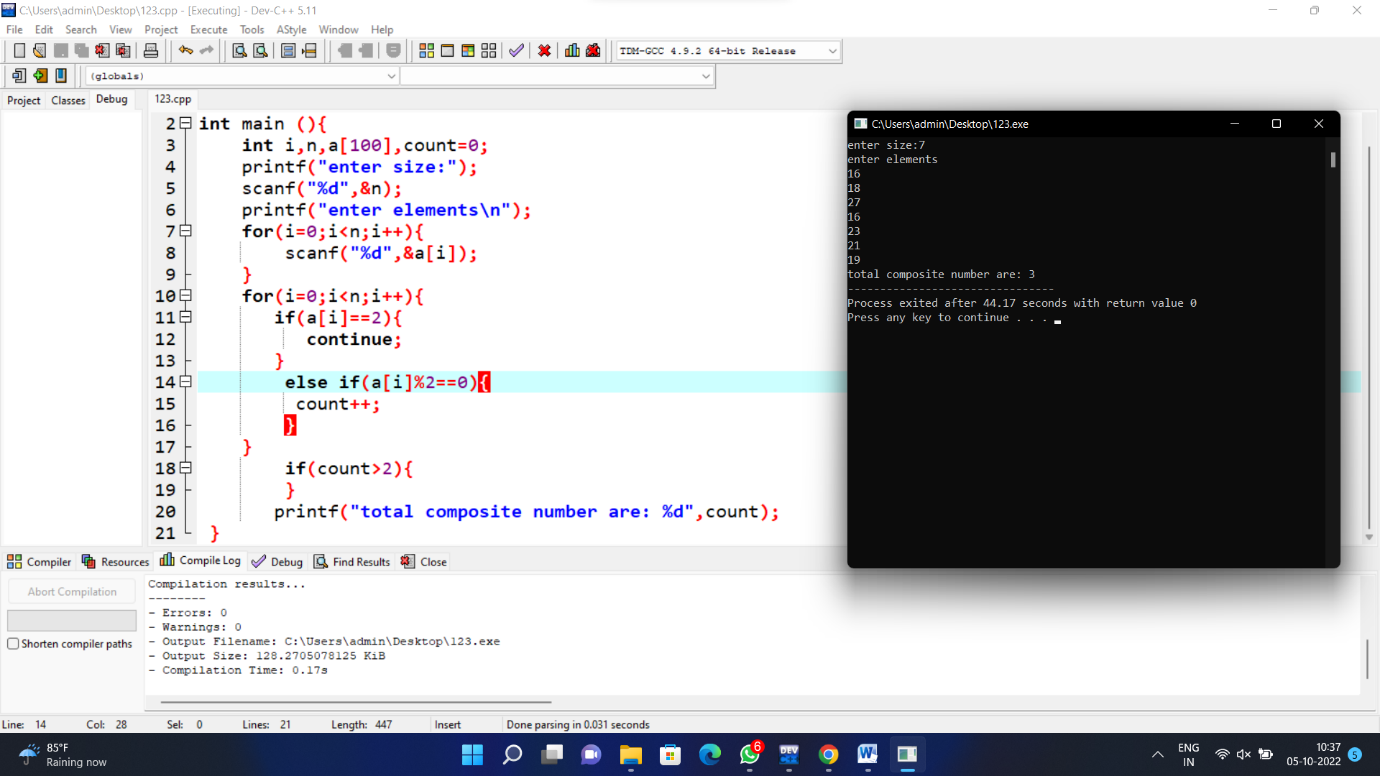
if(count>2){

}

printf("total composite number are: %d",count);

}

**OUTPUT**



2)NEGATIVE NUMBERS IN AN ARRAY

#include<stdio.h>

int main()

{

int Size, i, a[10];

printf("\n Please Enter the Size of an Array : ");

scanf("%d", &Size);

printf("\n Please Enter the Array Elements : ");

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

{

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

}

printf("\n List of Negative Numbers in this Array : ");

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

{

if(a[i] < 0)

{

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

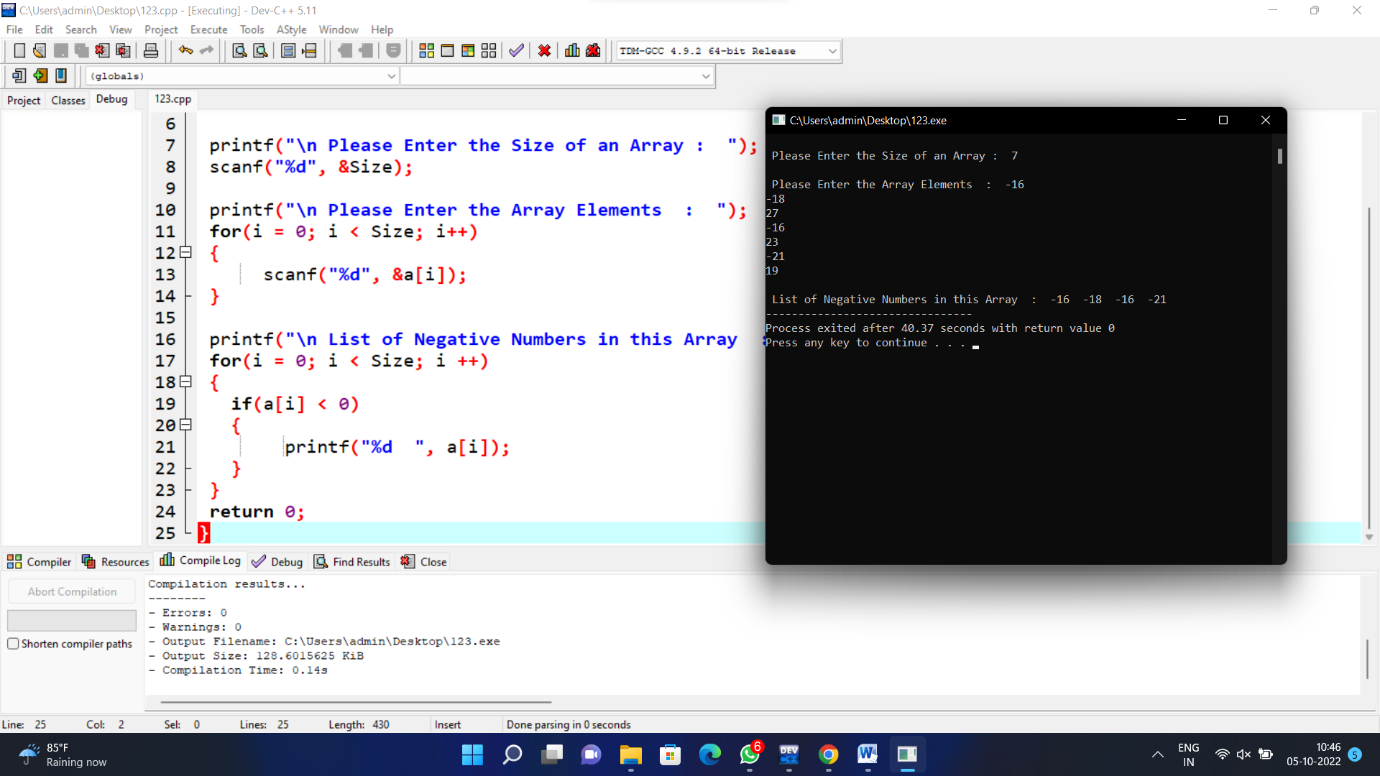
}

}

return 0;

}

**OUTPUT**



3)REVERSE AN ARRAY

#include <stdio.h>

int main()

{

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

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

printf("Original array: \n");

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

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

}

printf("\n");

printf("Array in reverse order: \n");

for (int i = length-1; i >= 0; i--) {

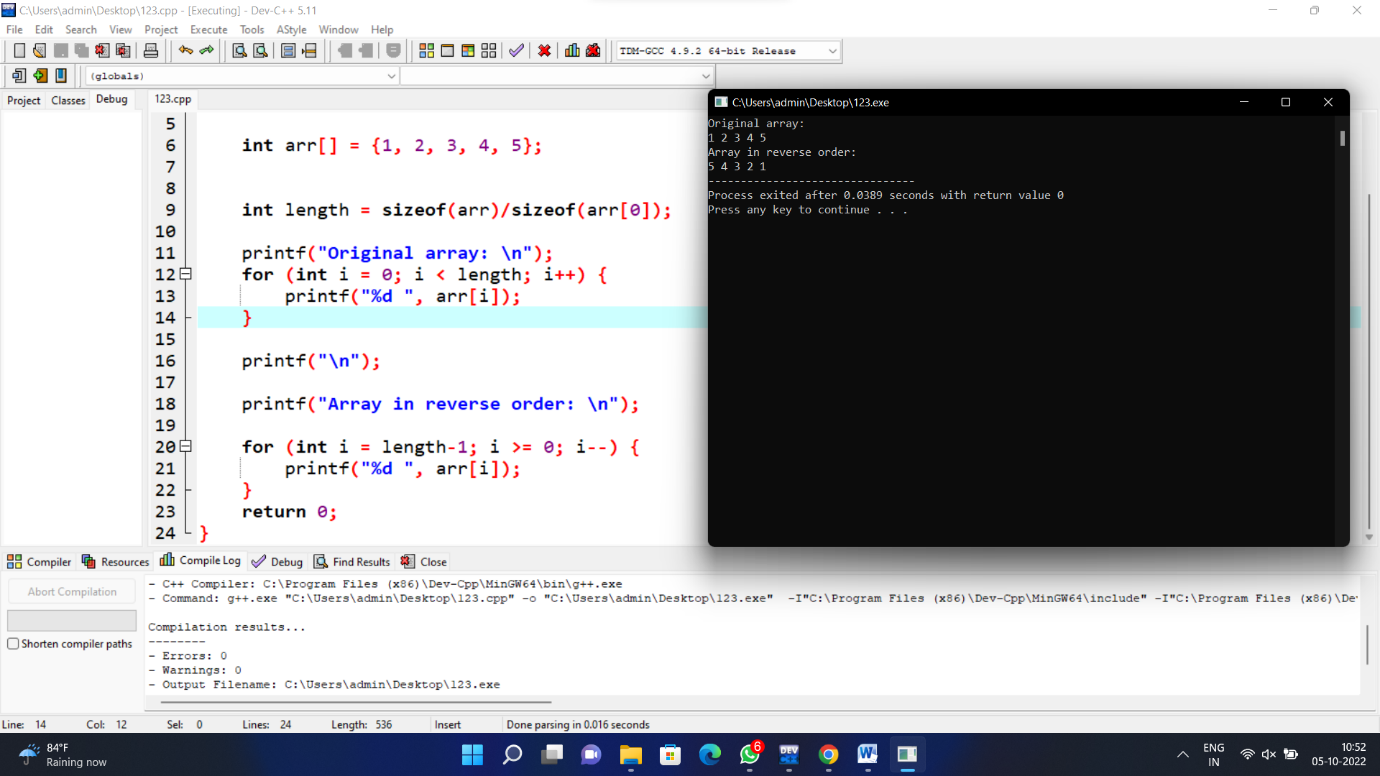
printf("%d ", arr[i]);

}

return 0;

}

**OUTPUT**



4)TRANSPOSE MATRIX

#include <stdio.h>

#define N 4

void transpose(int A[][N], int B[][N])

{

int i, j;

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

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

B[i][j] = A[j][i];

}

int main()

{

int A[N][N] = { { 1, 1, 1, 1 },

{ 2, 2, 2, 2 },

{ 3, 3, 3, 3 },

{ 4, 4, 4, 4 } };

int B[N][N], i, j;

transpose(A, B);

printf("Result matrix is \n");

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

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

printf("%d ", B[i][j]);

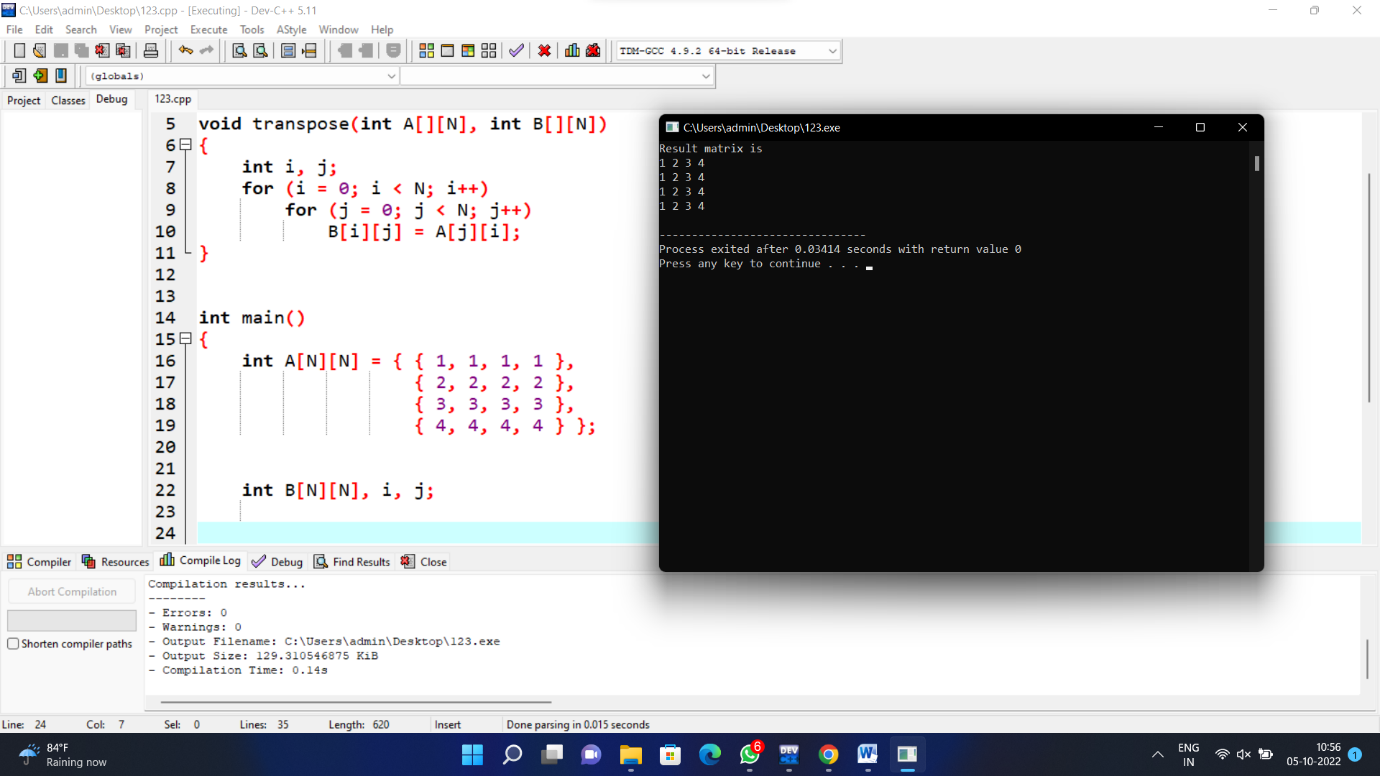
printf("\n");

}

return 0;

}

**OUTPUT**



5)REMOVE DUPLICATES

#include <stdio.h>

int removeDuplicates(int arr[], int n)

{

if (n == 0 || n == 1)

return n;

int temp[n];

int j = 0;

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

if (arr[i] != arr[i + 1])

temp[j++] = arr[i];

temp[j++] = arr[n - 1];

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

arr[i] = temp[i];

return j;

}

int main()

{

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

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

n = removeDuplicates(arr, n);

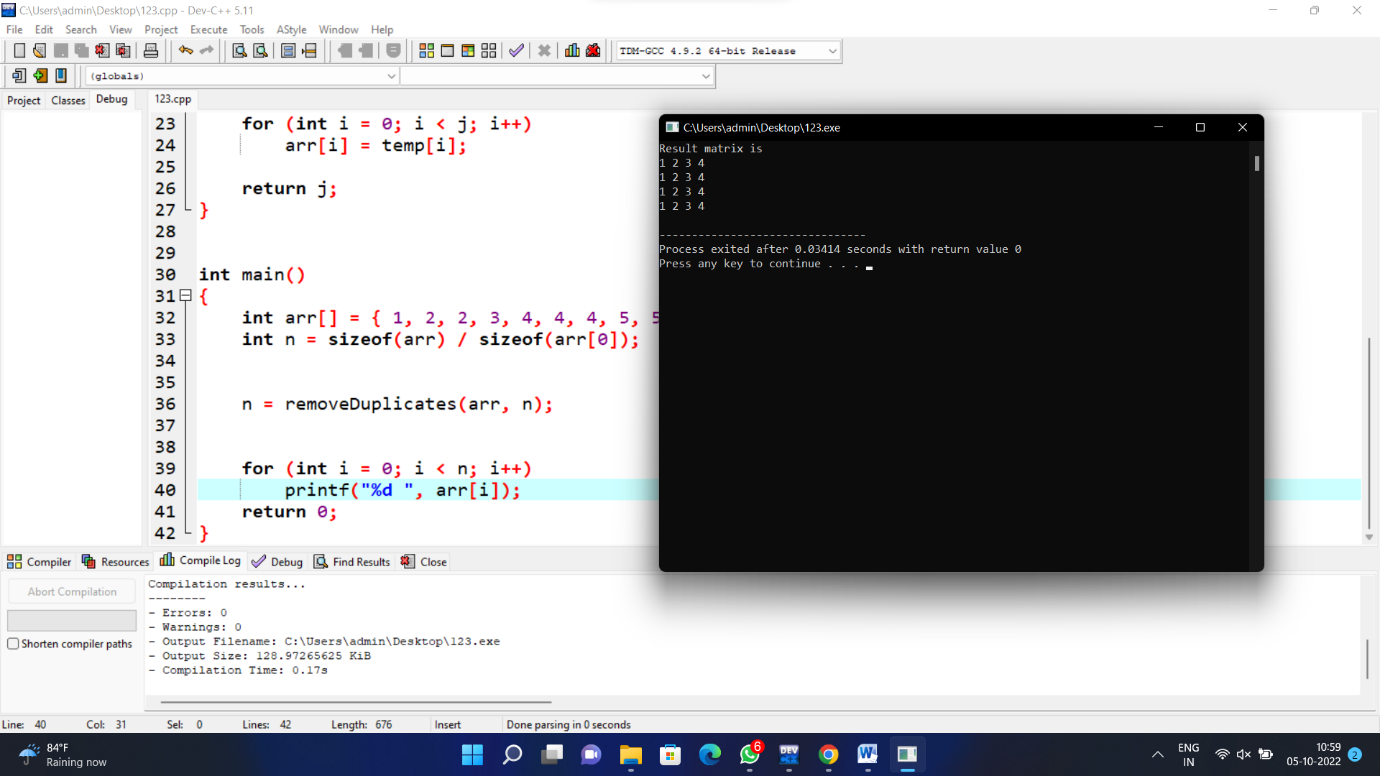
for (int i = 0; i < n; i++)

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

return 0;

}

**OUTPUT**



6)DIAGONAL MATRIX

#include <stdio.h>

const int M = 4;

const int N = 4;

void printDiagonalSums(int mat[M][N])

{

int principal = 0, secondary = 0;

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

{

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

{

if (i == j)

principal += mat[i][j];

if ((i + j) == (N - 1))

secondary += mat[i][j];

}

}

printf("%s", "Principal Diagonal:");

printf("%d\n", principal);

printf("%s", "Secondary Diagonal:");

printf("%d\n", secondary);

}

int main()

{

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

{5, 6, 7, 8},

{1, 2, 3, 4},

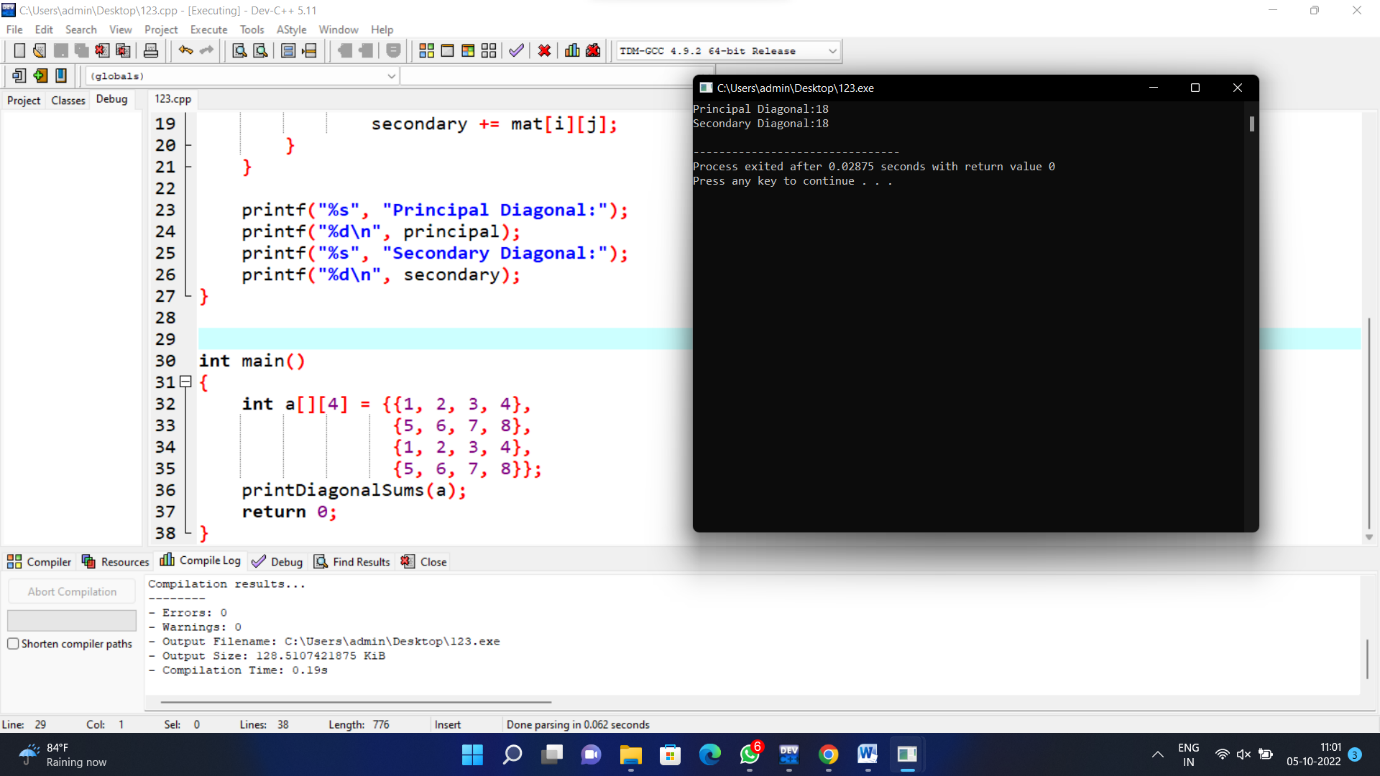
{5, 6, 7, 8}};

printDiagonalSums(a);

return 0;

}

**OUTPUT**



7)MINIMUM AND MAXIMUM

#include <stdio.h>

#include <conio.h>

int main()

{

int a[1000],i,n,min,max;

printf("Enter size of the array : ");

scanf("%d",&n);

printf("Enter elements in array : ");

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

{

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

}

min=max=a[0];

for(i=1; i<n; i++)

{

if(min>a[i])

min=a[i];

if(max<a[i])

max=a[i];

}

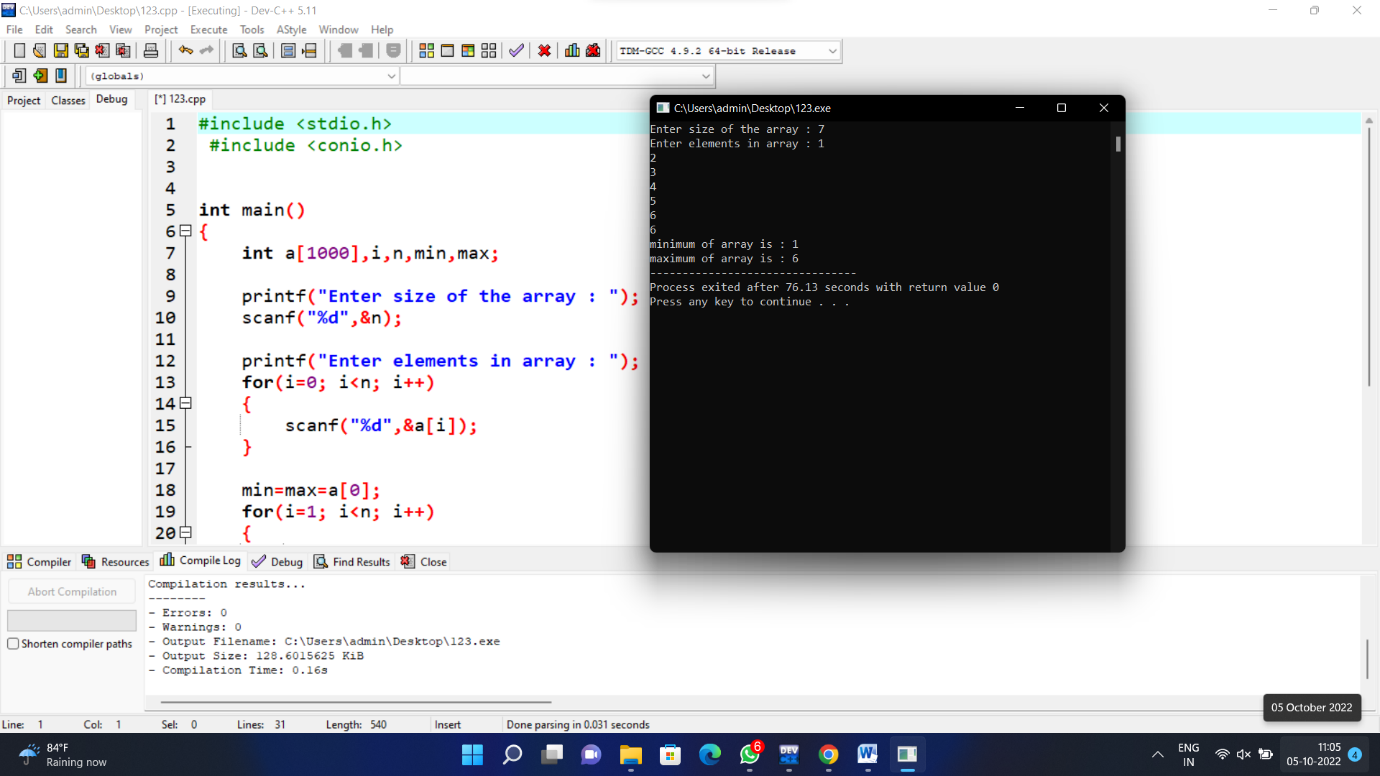
printf("minimum of array is : %d",min);

printf("\nmaximum of array is : %d",max);

return 0;

}

**OUTPUT**



8)NTH MAX & NTH MIN

#include <stdio.h>

int main()

{

int arr1[100];

int i, mx, mn, n;

printf("\n\nFind maximum and minimum element in an array :\n");

printf("--------------------------------------------------\n");

printf("Input the number of elements to be stored in the array :");

scanf("%d",&n);

printf("Input %d elements in the array :\n",n);

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

{

printf("element - %d : ",i);

scanf("%d",&arr1[i]);

}

mx = arr1[0];

mn = arr1[0];

for(i=1; i<n; i++)

{

if(arr1[i]>mx)

{

mx = arr1[i];

}

if(arr1[i]<mn)

{

mn = arr1[i];

}

}

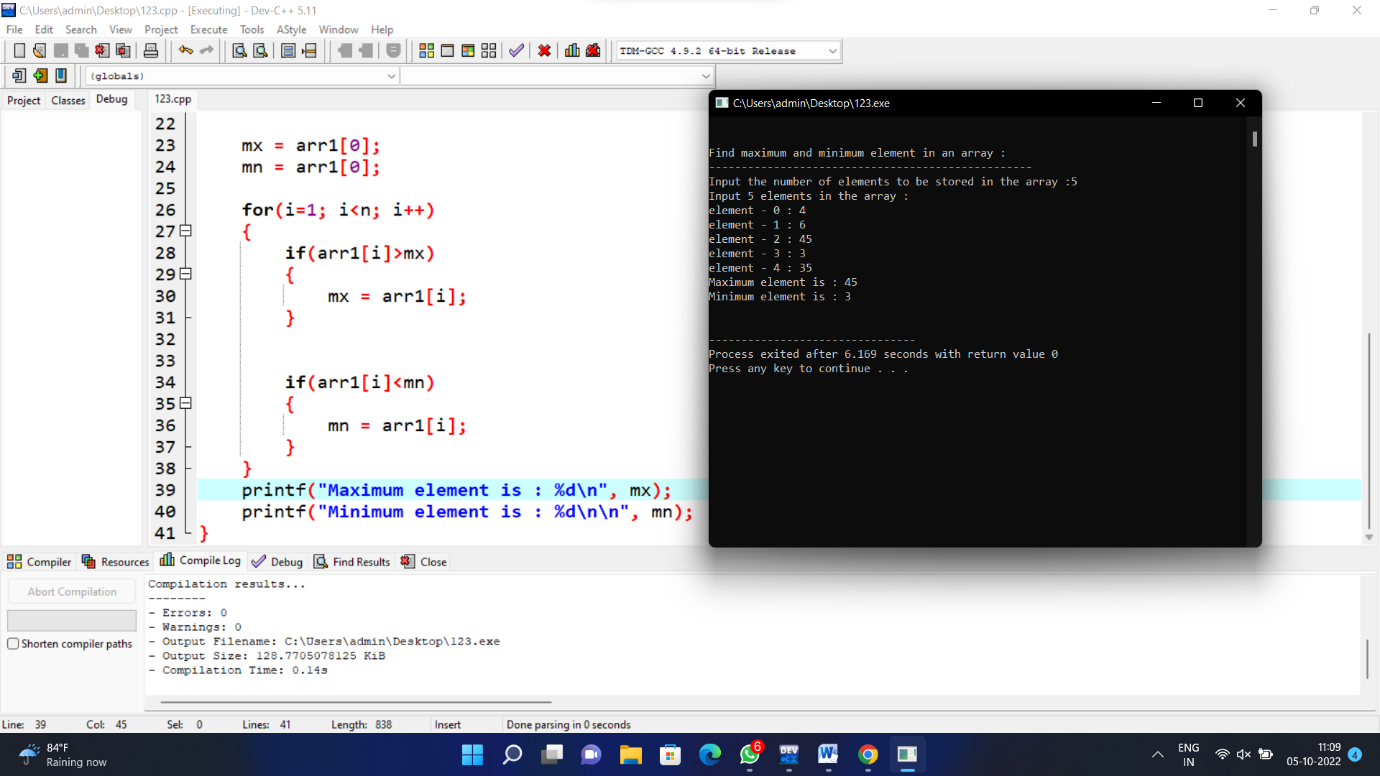
printf("Maximum element is : %d\n", mx);

printf("Minimum element is : %d\n\n", mn);

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**OUTPUT**



9)BINARY SEARCH

#include<stdio.h>

int main()

{

int c, first, last, middle, n, search, array[100];

printf("Enter number of elements\n");

scanf("%d",&n);

printf("Enter %d integers\n", n);

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

scanf("%d",&array[c]);

printf("Enter value to find\n");

scanf("%d",&search);

first = 0;

last = n - 1;

middle = (first+last)/2;

while( first <= last )

{

if ( array[middle] < search )

first = middle + 1;

else if ( array[middle] == search )

{

printf("%d found at location %d.\n", search, middle+1);

break;

}

else

last = middle - 1;

middle = (first + last)/2;

}

if ( first > last )

printf("Not found! %d is not present in the list.\n", search);

return 0;

}

**OUTPUT**

