***C PROGRAM LAB TASK 4 WITH SOLUTIONS***

1. Write a program in C to show the simple structure of a function.

#include <stdio.h>

int sum (int, int);//function declaration

int main ()

{

int total;

printf("\n\n Function : a simple structure of function :\n");

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

total = sum (21,171);//function call

printf ("The total is : %d\n", total);

return 0;

}

int sum (int a, int b) //function definition

{

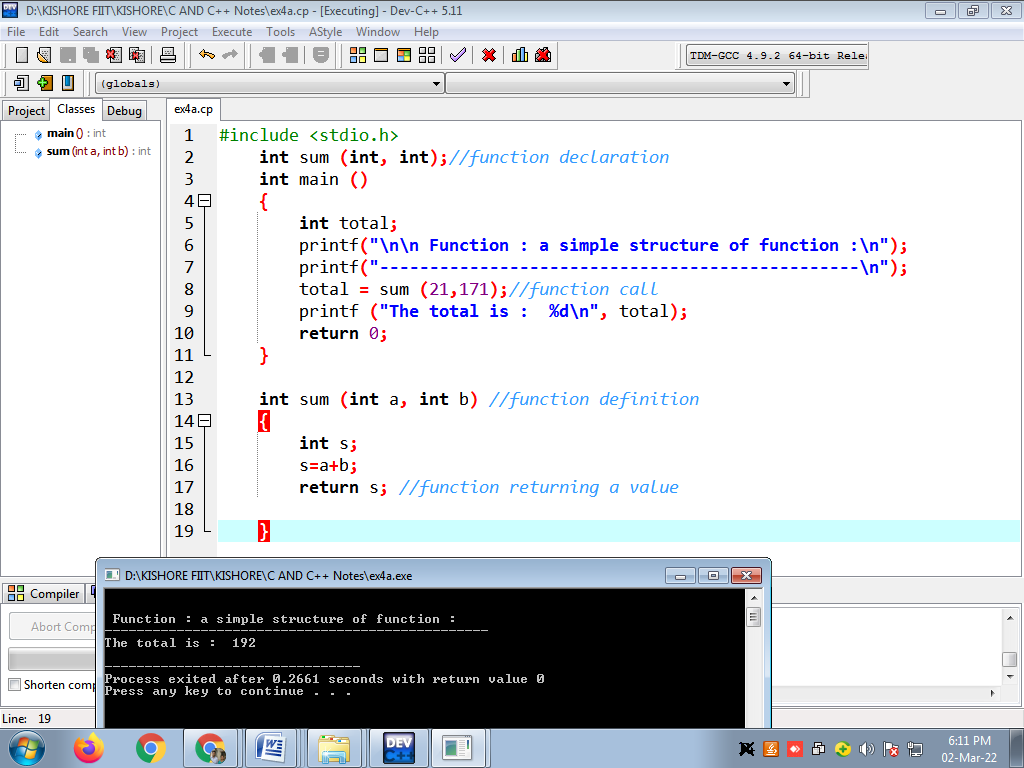
int s;

s=a+b;

return s; //function returning a value

}

Output:



**2.Write a program in C to find the square of any number using the function.**

#include <stdio.h>

double square(double num)

{

return (num \* num);

}

int main()

{

int num;

double n;

printf("\n\n Function : find square of any number :\n");

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

printf("Input any number for square : ");

scanf("%d", &num);

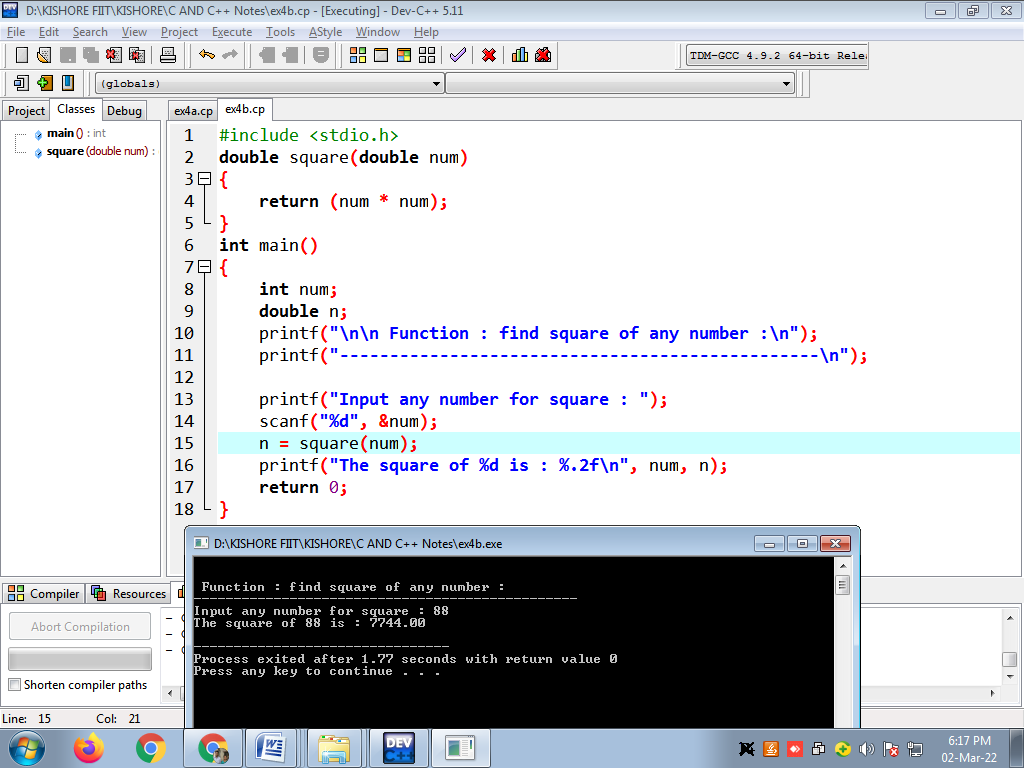
n = square(num);

printf("The square of %d is : %.2f\n", num, n);

return 0;

}

Output:

**3.**Write a program in C to swap two numbers using function.

#include<stdio.h>

void swap(int \*,int \*);

int main()

{

int n1,n2;

printf("\n\n Function : swap two numbers using function :\n");

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

printf("Input 1st number : ");

scanf("%d",&n1);

printf("Input 2nd number : ");

scanf("%d",&n2);

printf("Before swapping: n1 = %d, n2 = %d ",n1,n2);

swap(&n1,&n2);

printf("\nAfter swapping: n1 = %d, n2 = %d \n\n",n1,n2);

return 0;

}

void swap(int \*p,int \*q)

{

int tmp;

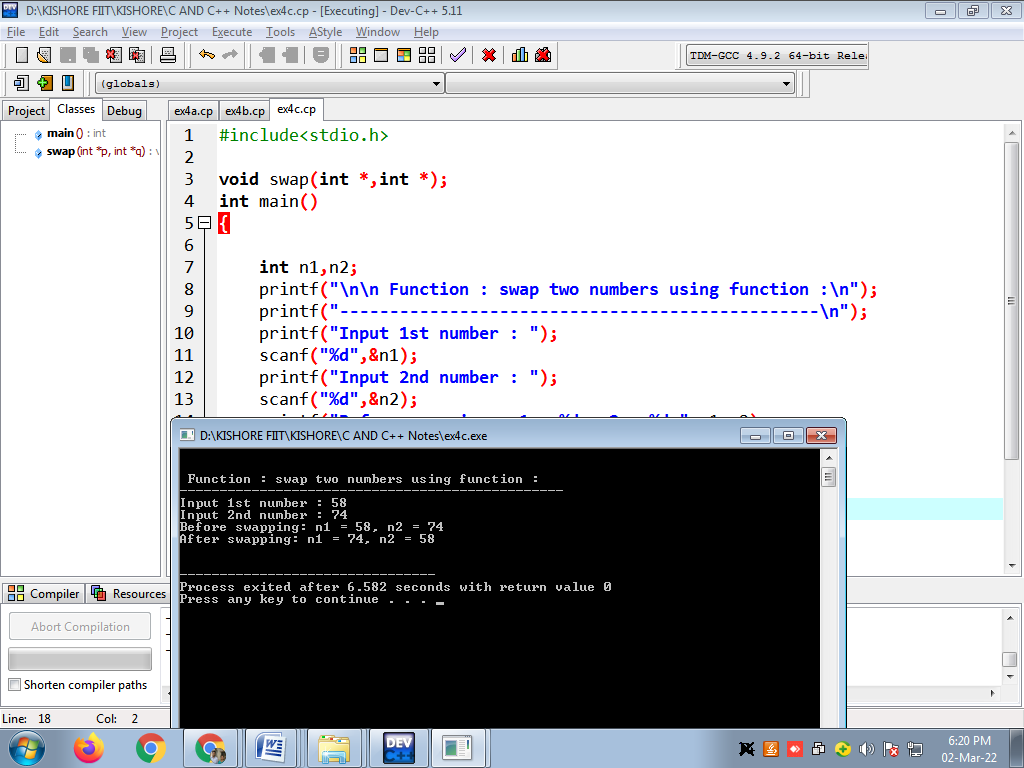
tmp = \*p;

\*p=\*q;

\*q=tmp;

}

Output:



4. Write a program in C to check a given number is even or odd using the function.

#include <stdio.h>

int checkOddEven(int n1)

{

return (n1 & 1);

}

int main()

{

int n1;

printf("\n\n Function : check the number is even or odd:\n");

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

printf("Input any number : ");

scanf("%d", &n1);

if(checkOddEven(n1))

{

printf("The entered number is odd.\n\n");

}

else

{

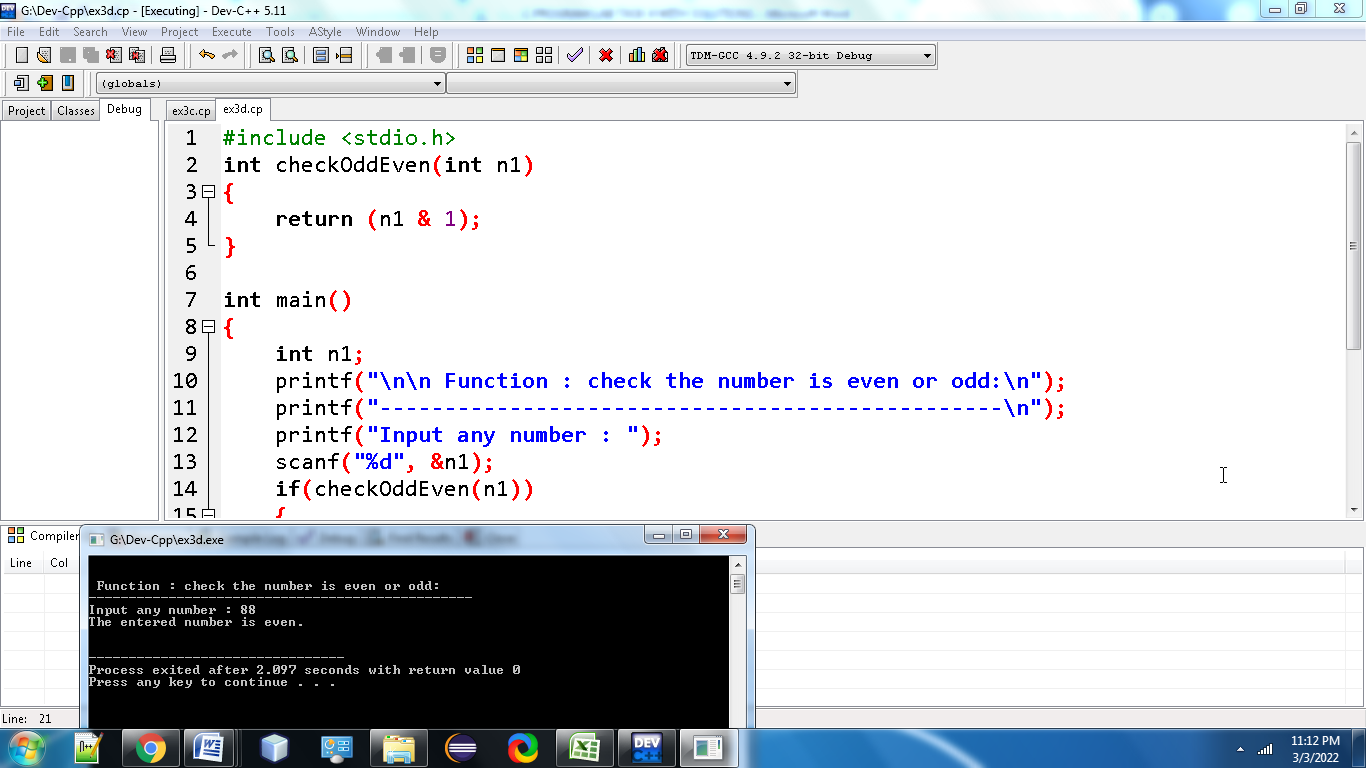
printf("The entered number is even.\n\n");

}

return 0;

}

**OUTPUT:**

****

5. Write a program in C to find the sum of the series 1!/1+2!/2+3!/3+4!/4+5!/5 using the function.

**#include <stdio.h>**

**int fact(int);**

**int main()**

**{**

**int sum;**

**sum=fact(1)/1+fact(2)/2+fact(3)/3+fact(4)/4+fact(5)/5;**

**printf("\n\n Function : find the sum of 1!/1+2!/2+3!/3+4!/4+5!/5 :\n");**

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

**printf("The sum of the series is : %d\n\n",sum);**

**return 0;**

**}**

**int fact(int n)**

**{**

**int num=0,f=1;**

**while(num<=n-1)**

**{**

**f =f+f\*num;**

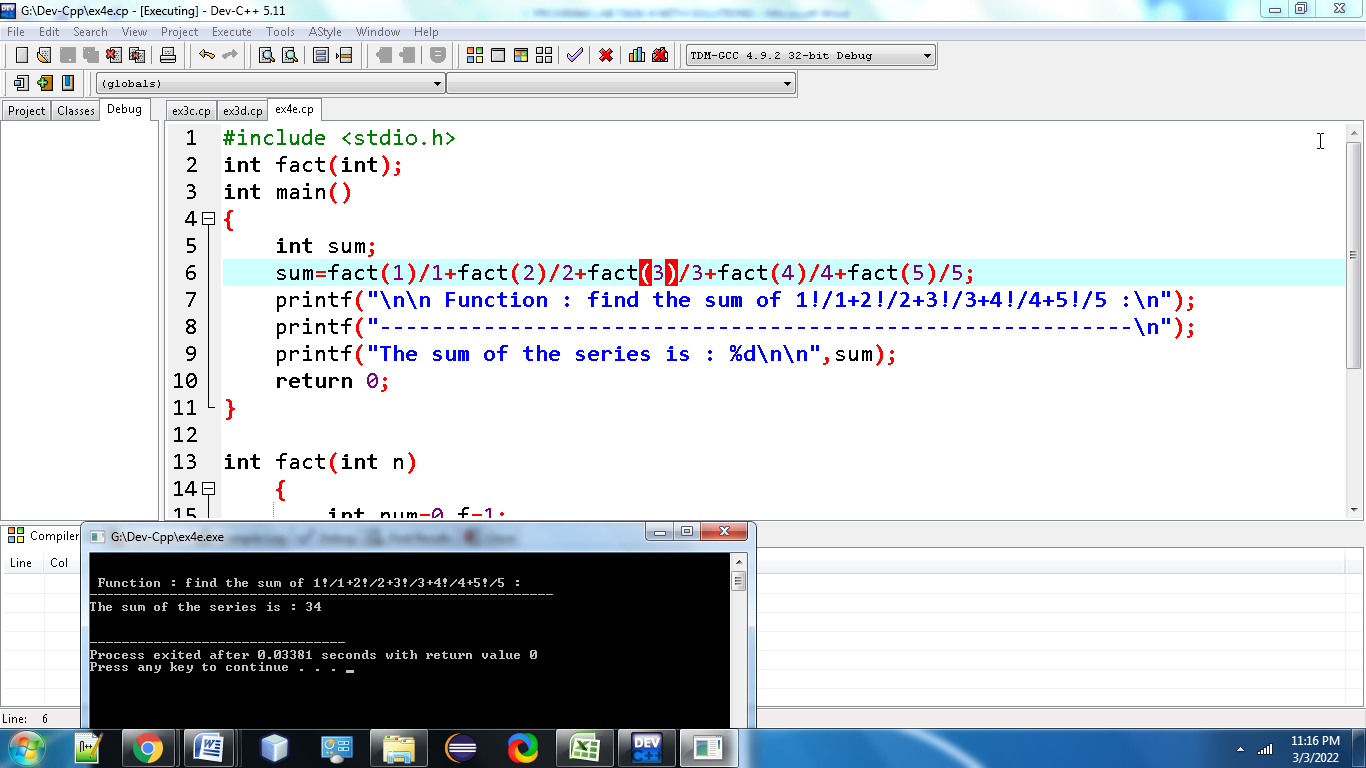
**num++;**

**}**

**return f;**

**}**

**Output:**

****

6. Write a program in C to convert decimal number to binary number using the function.

**#include<stdio.h>**

**long toBin(int);**

**int main()**

**{**

**long bno;**

**int dno;**

**printf("\n\n Function : convert decimal to binary :\n");**

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

**printf(" Input any decimal number : ");**

**scanf("%d",&dno);**

**bno = toBin(dno);**

**printf("\n The Binary value is : %ld\n\n",bno);**

**return 0;**

**}**

**long toBin(int dno)**

**{**

**long bno=0,remainder,f=1;**

**while(dno != 0)**

**{**

**remainder = dno % 2;**

**bno = bno + remainder \* f;**

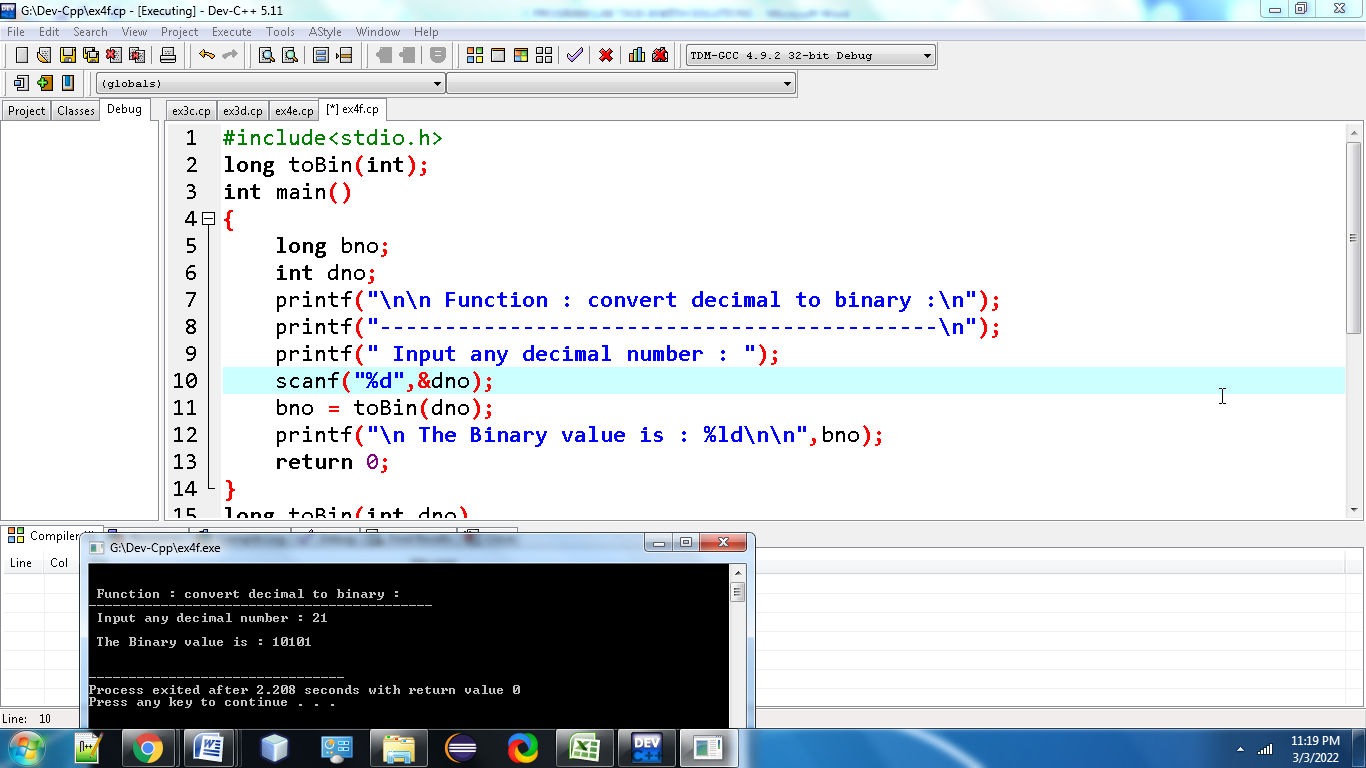
**f = f \* 10;**

**dno = dno / 2;**

**}**

**return bno;**

**}**

**Output: **

**7.**Write a program in C to check whether a number is a prime number or not using the function.

#include<stdio.h>

int PrimeOrNot(int);

int main()

{

int n1,prime;

printf("\n\n Function : check whether a number is prime number or not :\n");

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

printf(" Input a positive number : ");

scanf("%d",&n1);

prime = PrimeOrNot(n1);

if(prime==1)

printf(" The number %d is a prime number.\n",n1);

else

printf(" The number %d is not a prime number.\n",n1);

return 0;

}

int PrimeOrNot(int n1)

{

int i=2;

while(i<=n1/2)

{

if(n1%i==0)

return 0;

else

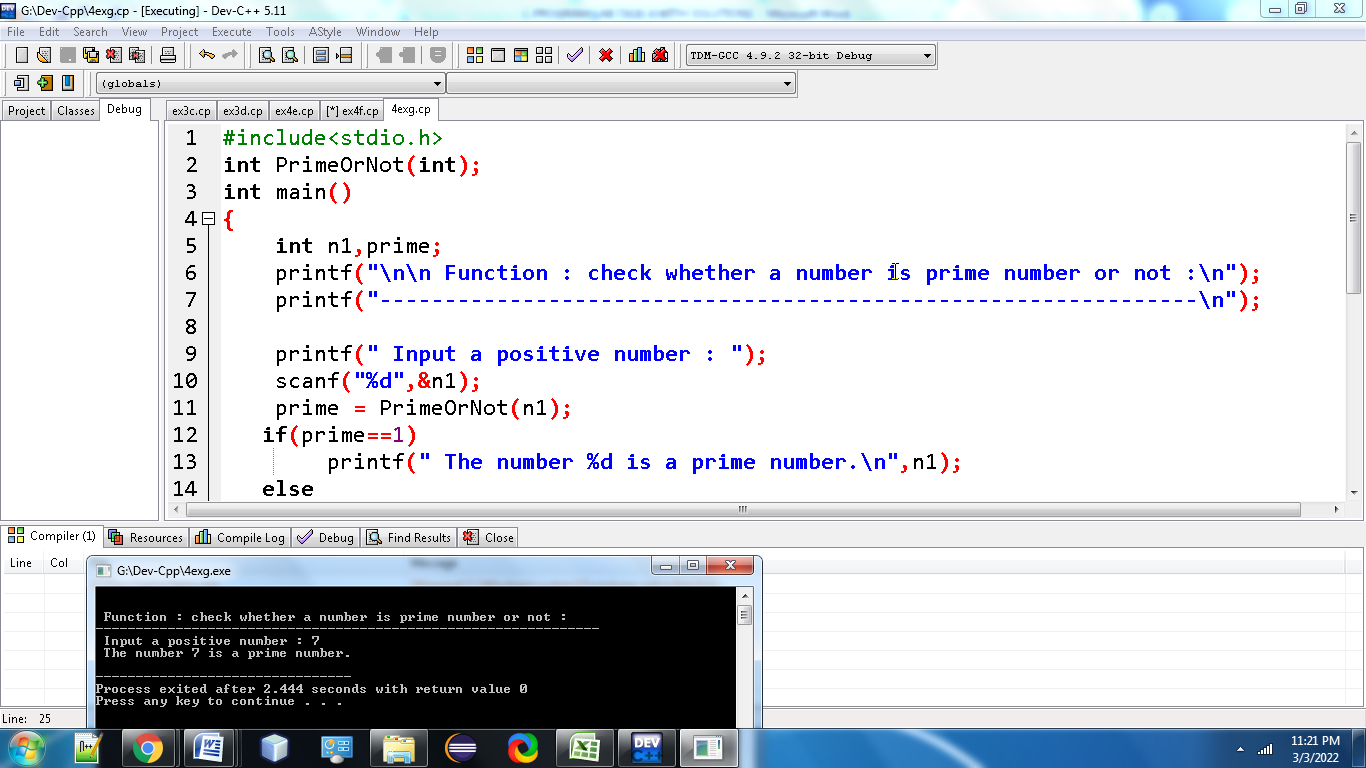
i++;

}

return 1;

}

Output:



**8.**Write a program in C to get the largest element of an array using the function.

#include<stdio.h>

#define MAX 100

int findMaxElem(int []);

int n;

int main()

{

int arr1[MAX],mxelem,i;

printf("\n\n Function : get largest element of 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]);

}

mxelem=findMaxElem(arr1);

printf(" The largest element in the array is : %d\n\n",mxelem);

return 0;

}

int findMaxElem(int arr1[])

{

int i=1,mxelem;

mxelem=arr1[0];

while(i < n)

{

if(mxelem<arr1[i])

mxelem=arr1[i];

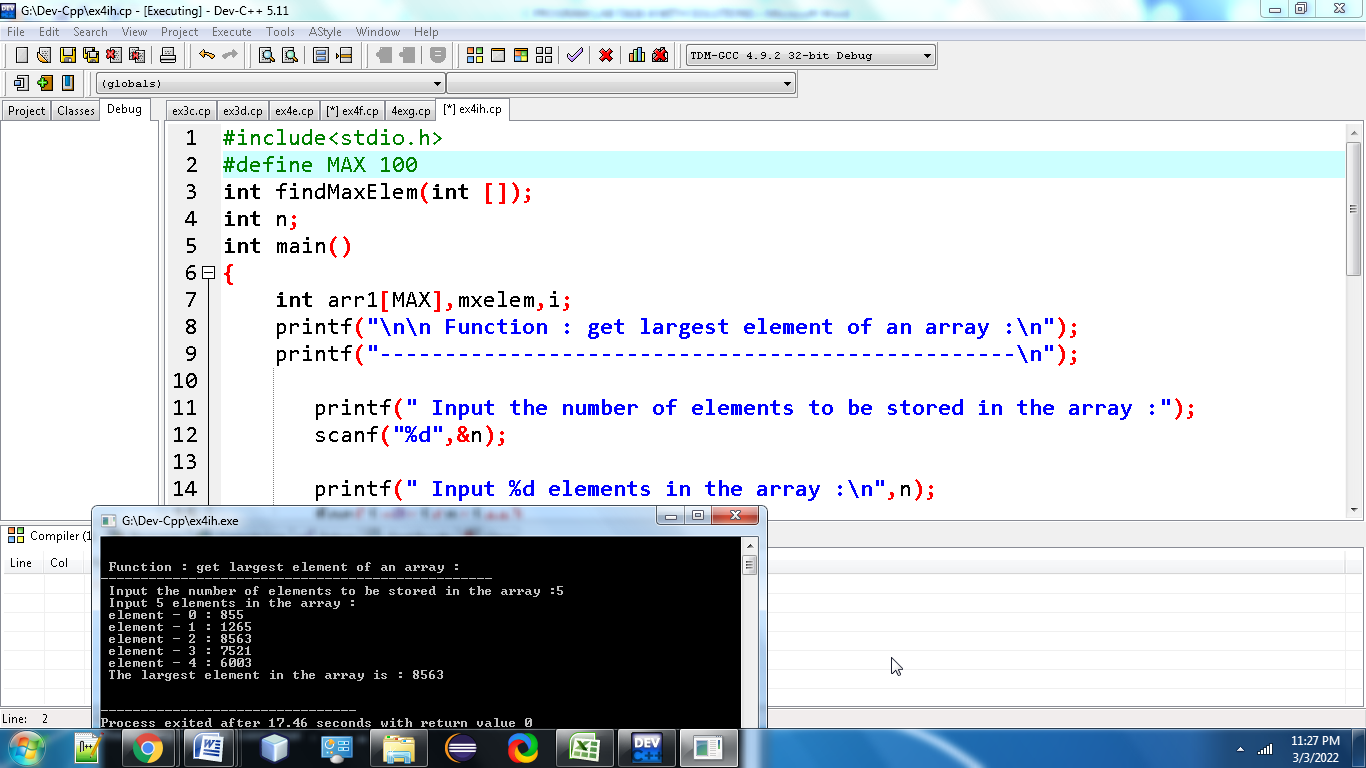
i++;

}

return mxelem;

}

Output:



**9. Write a program in C to check Armstrong and perfect numbers using the function.**

#include <stdio.h>

int checkArmstrong(int n1);

int checkPerfect(int n1);

int main()

{

int n1;

printf("\n\n Function : check Armstrong and perfect numbers :\n");

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

printf(" Input any number: ");

scanf("%d", &n1);

if(checkArmstrong(n1))

{

printf(" The %d is an Armstrong number.\n", n1);

}

else

{

printf(" The %d is not an Armstrong number.\n", n1);

}

if(checkPerfect(n1))

{

printf(" The %d is a Perfect number.\n\n", n1);

}

else

{

printf(" The %d is not a Perfect number.\n\n", n1);

}

return 0;

}

int checkArmstrong(int n1)

{

int ld, sum, num;

sum = 0;

num = n1;

while(num!=0)

{

ld = num % 10;

sum += ld \* ld \* ld;

num = num/10;

}

return (n1 == sum);

}

int checkPerfect(int n1)

{

int i, sum, num;

sum = 0;

num = n1;

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

{

if(num%i == 0)

{

sum += i;

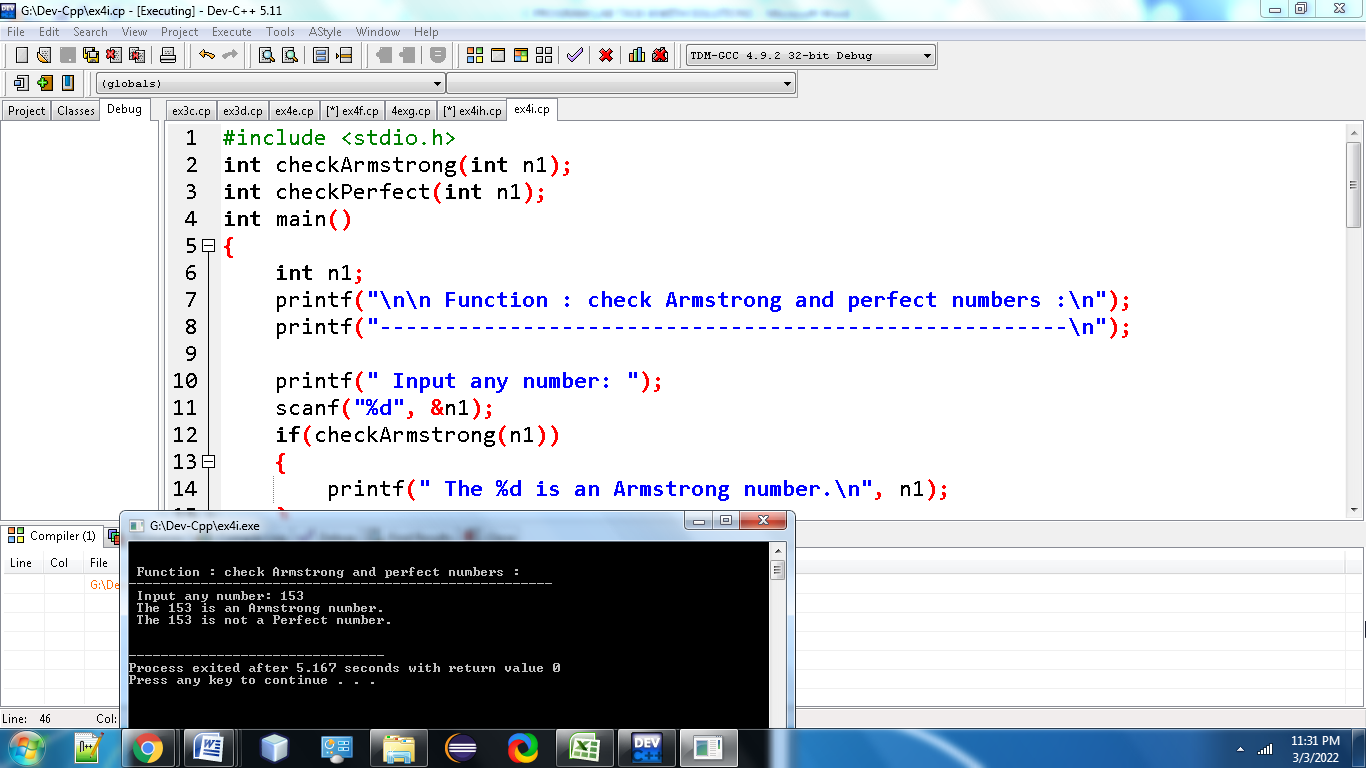
}

}

return (n1 == sum);

}

OUTPUT:



10. Write a program in C to print all perfect numbers in given range using the function.

#include <stdio.h>

int checkPerfect(int n1);

void PerfectNumbers(int stLimit, int enLimit);

int main()

{

int stLimit, enLimit;

printf("\n\n Function : perfect numbers in a given range :\n");

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

printf(" Input lowest search limit of perfect numbers : ");

scanf("%d", &stLimit);

printf(" Input highest search limit of perfect numbers : ");

scanf("%d", &enLimit);

printf("\n The perfect numbers between %d to %d are : \n", stLimit, enLimit);

PerfectNumbers(stLimit, enLimit);

printf("\n\n");

return 0;

}

int checkPerfect(int n1)

{

int i, sum;

sum = 0;

for(i=1; i<n1; i++)

{

if(n1 % i == 0)

{

sum += i;

}

}

if(sum == n1)

return 1;

else

return 0;

}

void PerfectNumbers(int stLimit, int enLimit)

{

while(stLimit <= enLimit)

{

if(checkPerfect(stLimit))

{

printf(" %d ", stLimit);

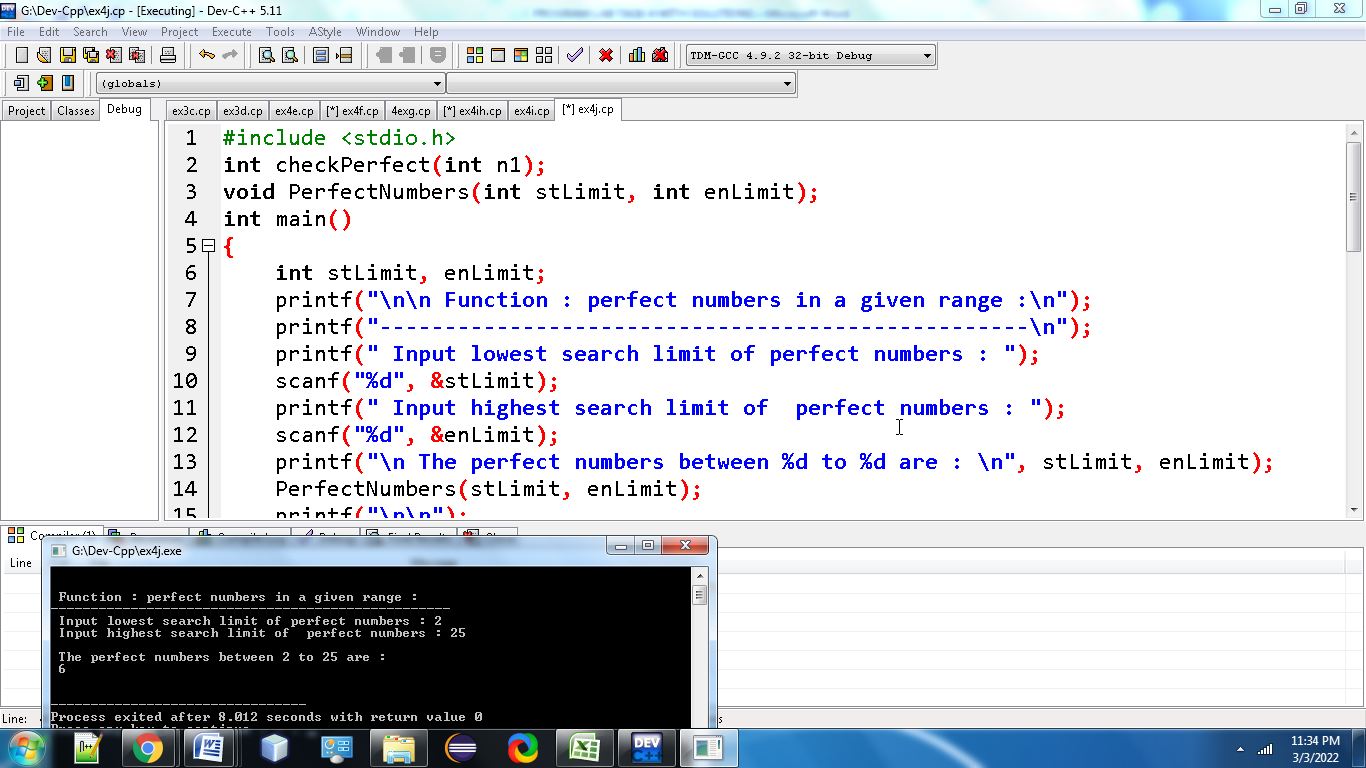
}

stLimit++;

}

}

OUTPUT:



**11. Write a program in C to check whether two given strings are an anagram.**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int checkAnagram(char \*str1, char \*str2);

int main()

{

char str1[100], str2[100];

printf("\n\n Function : whether two given strings are anagram :\n");

printf("\n\n Example : pears and spare, stone and tones :\n");

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

printf(" Input the first String : ");

fgets(str1, sizeof str1, stdin);

printf(" Input the second String : ");

fgets(str2, sizeof str2, stdin);

if(checkAnagram(str1, str2) == 1)

{

str1[strlen(str1)-1] = '\0';

str2[strlen(str2)-1] = '\0';

printf(" %s and %s are Anagram.\n\n",str1,str2);

}

else

{

str1[strlen(str1)-1] = '\0';

str2[strlen(str2)-1] = '\0';

printf(" %s and %s are not Anagram.\n\n",str1,str2);

}

return 0;

}

int checkAnagram(char \*str1, char \*str2)

{

int str1ChrCtr[256] = {0}, str2ChrCtr[256] = {0};

int ctr;

if(strlen(str1) != strlen(str2))

{

return 0;

}

for(ctr = 0; str1[ctr] != '\0'; ctr++)

{

str1ChrCtr[str1[ctr]]++;

}

for(ctr = 0; str2[ctr] != '\0'; ctr++)

{

str2ChrCtr[str2[ctr]]++;

}

for(ctr = 0; ctr < 256; ctr++)

{

if(str1ChrCtr[ctr] != str2ChrCtr[ctr])

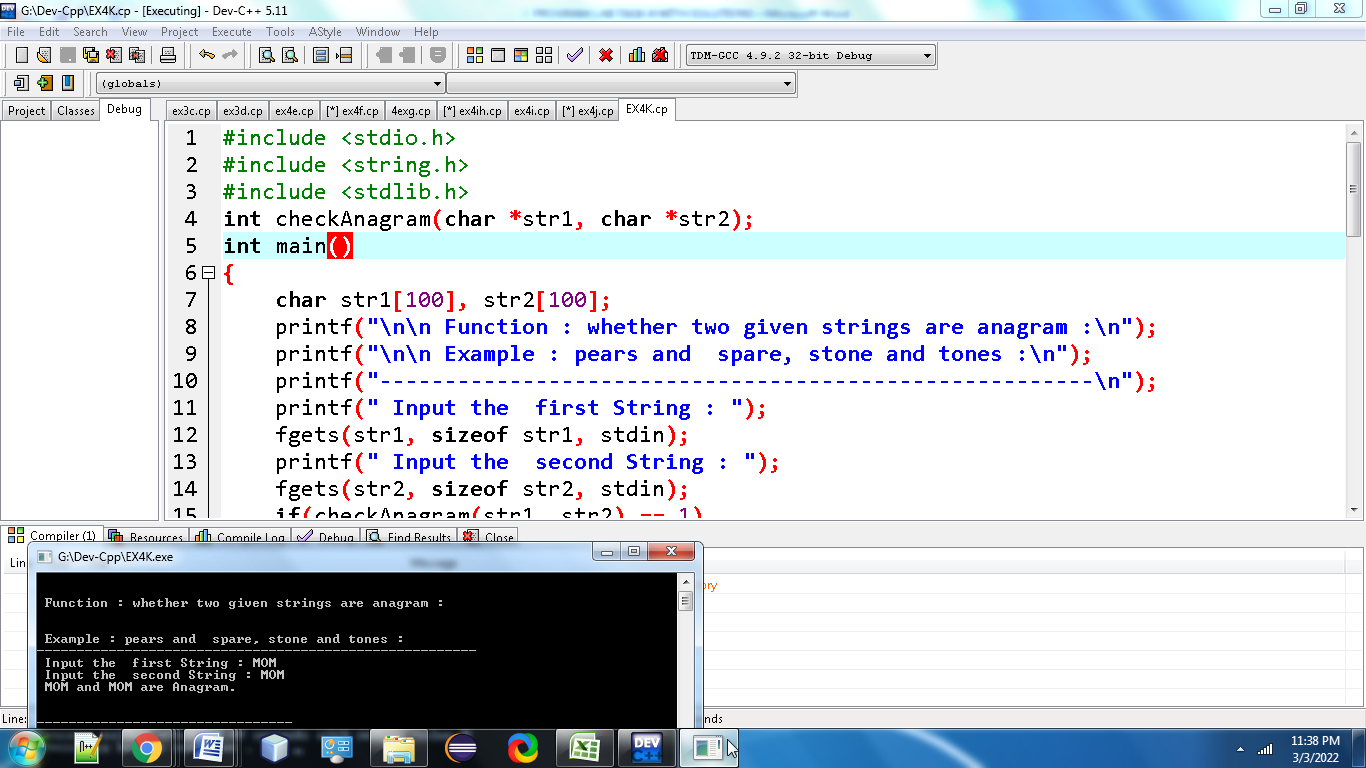
return 0;

}

return 1;

}

OUTPUT:



12. Write a C programming to find out maximum and minimum of some values using function which will return an array.

# include <stdio.h>

# define max 10

int \*maxmin(int ar[], int v);

int main()

{

int arr[max];

int n,i, \*p;

printf("Number of values you want to input: ");

scanf("%d",&n);

printf("Input %d values\n", n);

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

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

p=maxmin(arr,n);

printf("Minimum value is: %d\n",\*p++);

printf("Maximum value is: %d\n",\*p);

}

int \*maxmin(int arra1[], int v)

{

int i;

static int result\_mm[2];

result\_mm[0]=arra1[0];

result\_mm[1]=arra1[0];

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

{

if(result\_mm[0] > arra1[i])

result\_mm[0]=arra1[i];

if(result\_mm[1]< arra1[i])

result\_mm[1]= arra1[i];

}

return result\_mm;

}

Output:

