CSA0396 DATA STRUCTURE (PRACTICAL).

Day 1

IDENTIFY EVEN OR ODD IN ARRAY SET?

CODE :

#include<stdio.h>

int main()

{

int array[100],i,num;

printf("enter the size of array\n");

scanf("%d",&num);

printf("enter the array element of array\n");

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

{

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

}

printf("even number in the array are ");

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

{

if (array[i]%2 == 0)

{

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

}

}

printf("odd number in the array are");

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

{

if (array[i]%2 != 0)

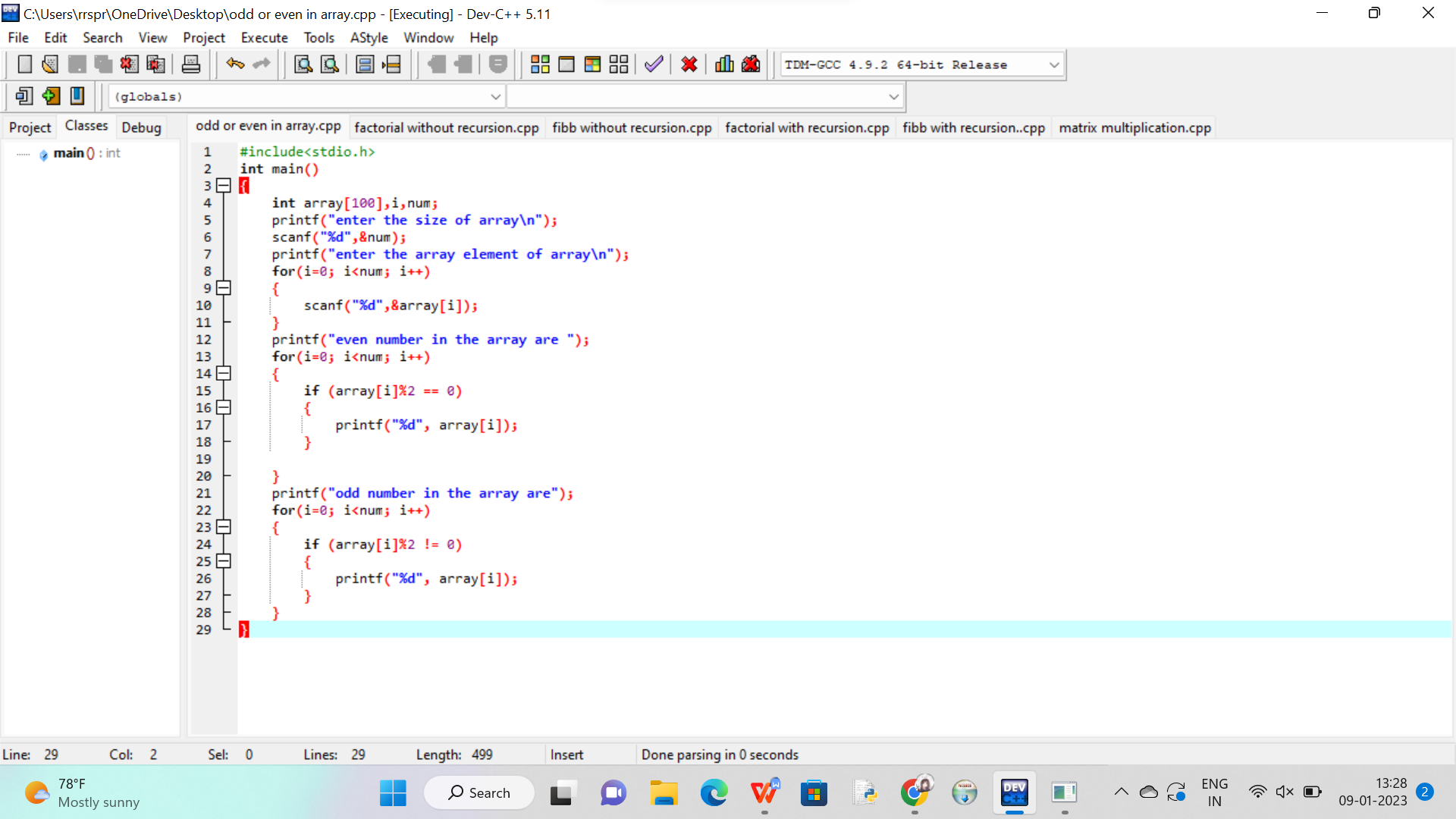
{

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

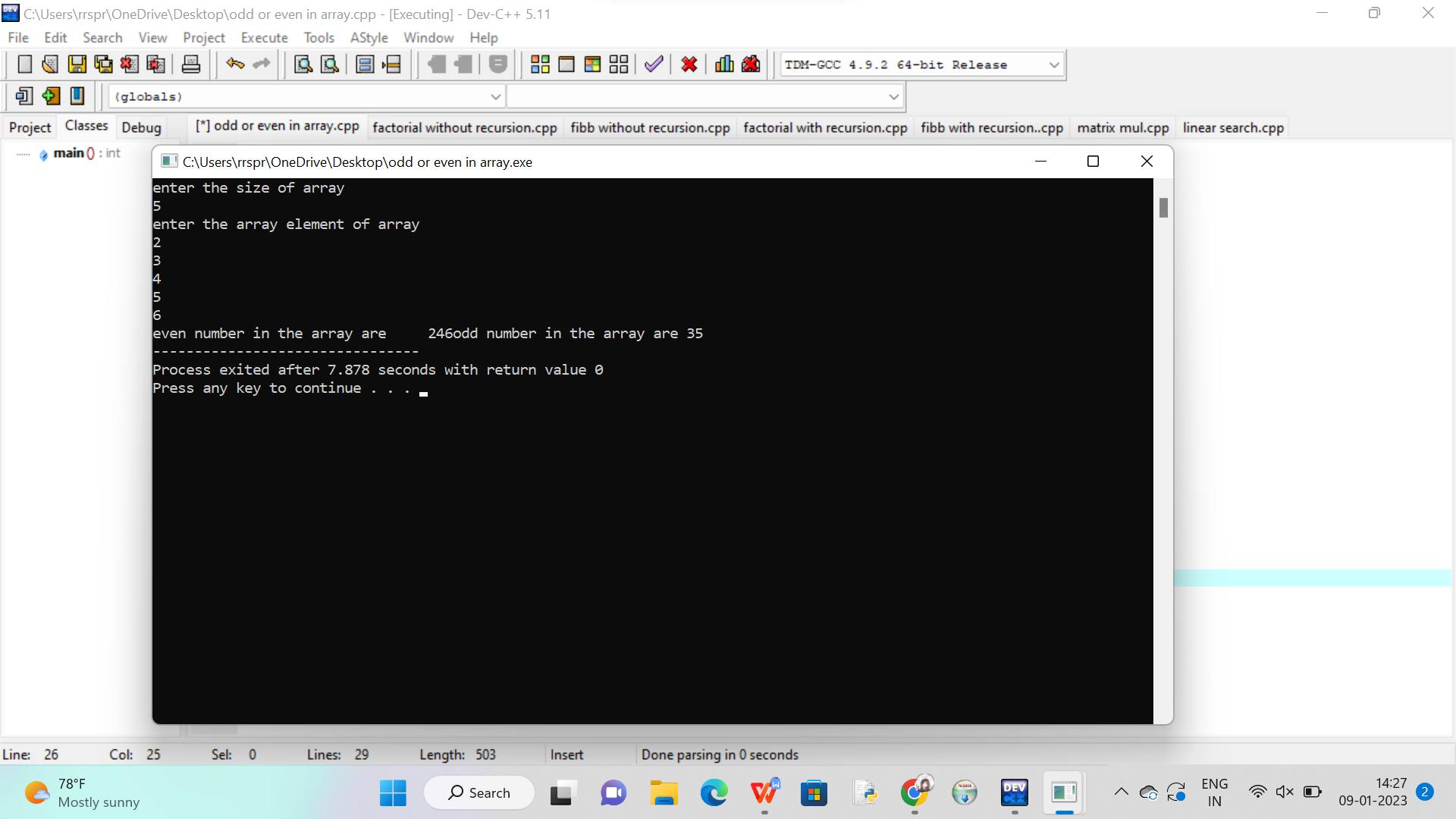
}

}

}



Out put:



FACTORIAL WITH RECURSION :

CODE :

#include<stdio.h>

int fact(int n)

{

if (n!=0)

return n\*fact(n-1);

else

return 1;

}

int main()

{

int num,result;

printf("enter a factorial number:");

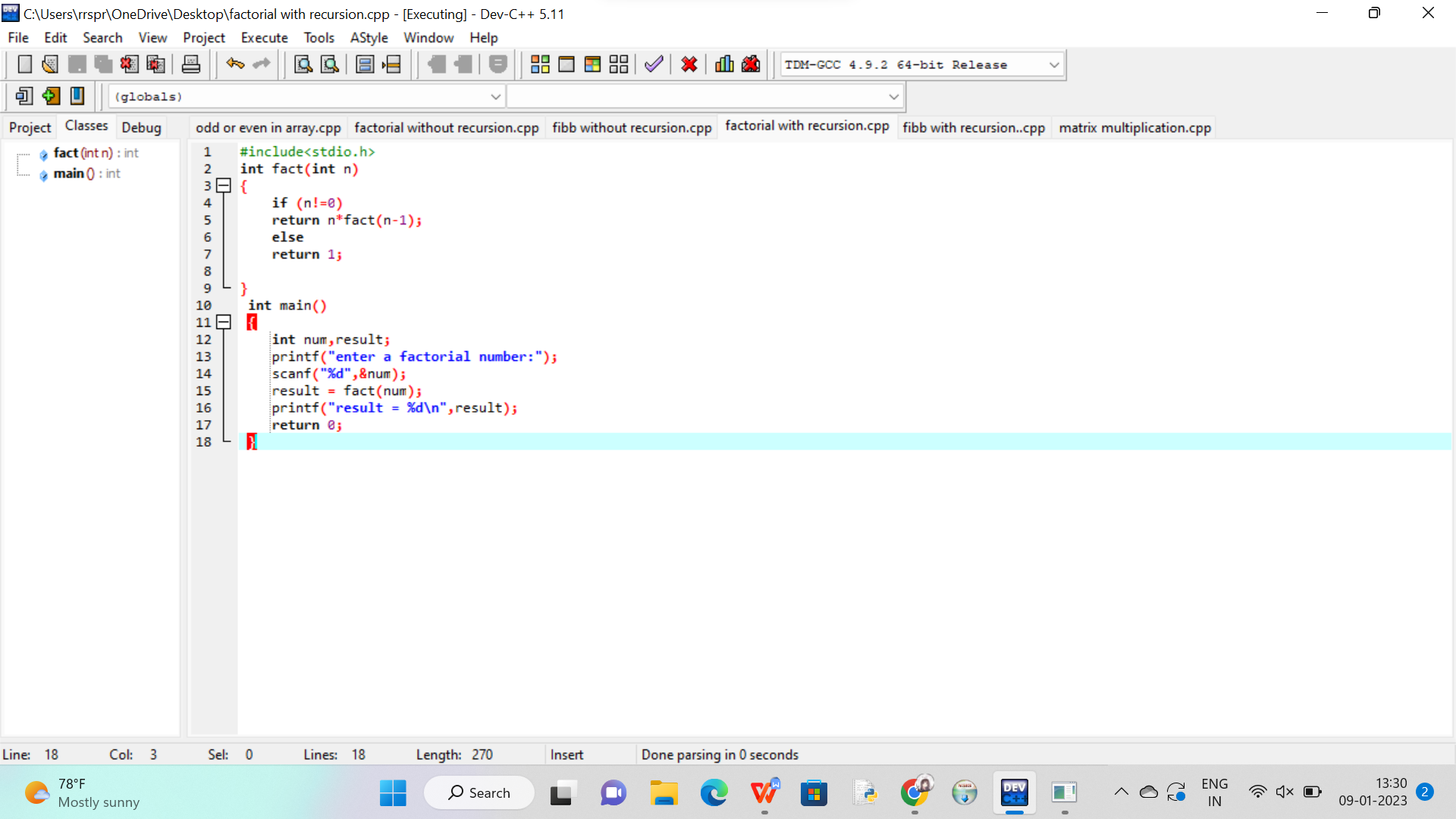
scanf("%d",&num);

result = fact(num);

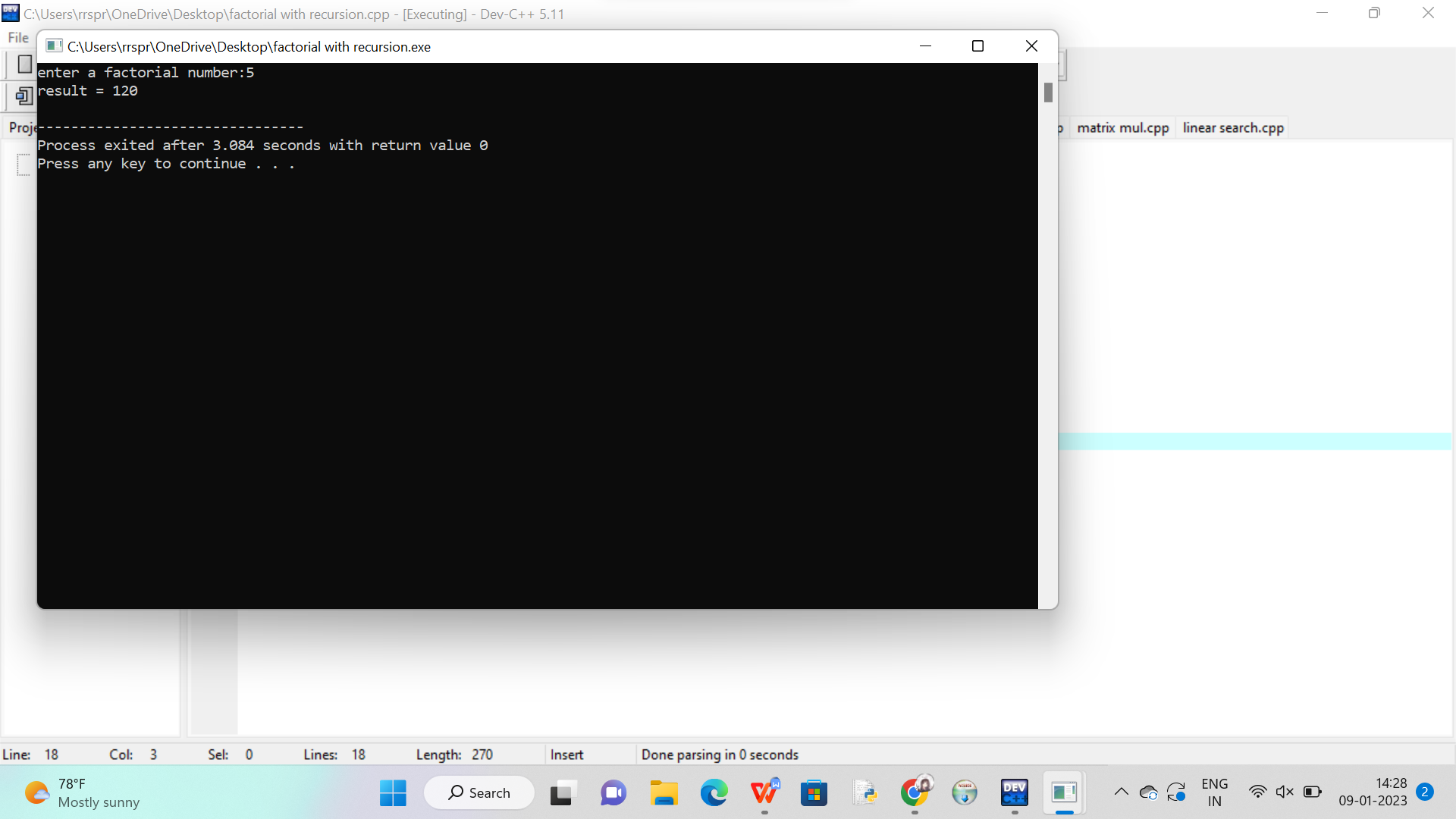
printf("result = %d\n",result);

return 0;

}



OUTPUT:



FACTORIAL WITHOUT RECURSION :

CODE :

#include<stdio.h>

int main()

{

int i,f=1,num;

printf ("enter the number:");

scanf("%d",&num);

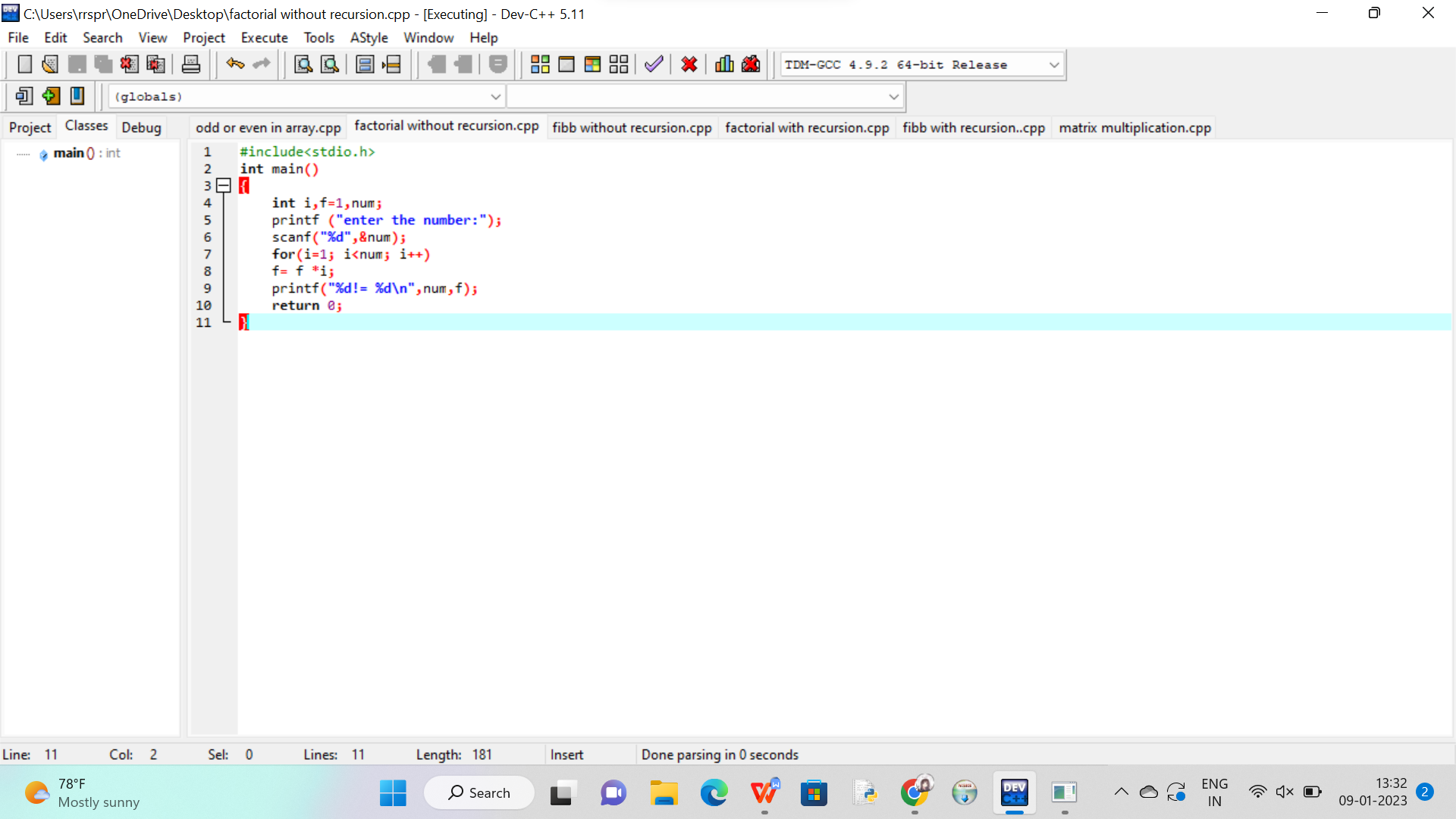
for(i=1; i<num; i++)

f= f \*i;

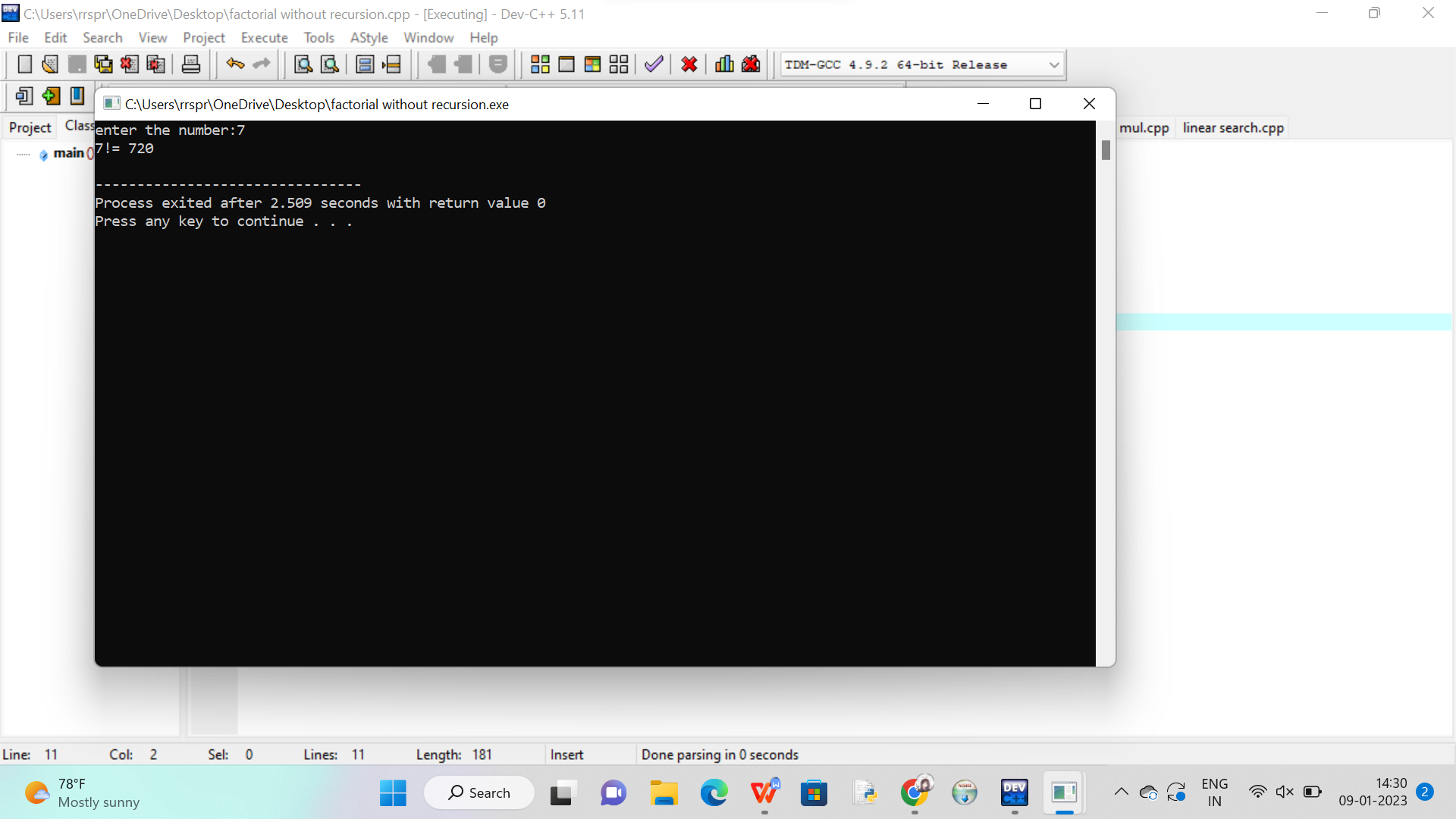
printf("%d!= %d\n",num,f);

return 0;

}



OUTPUT:



FIBBONACI SERIES WITHOUT RECURSION :

CODE :

#include<stdio.h>

int main()

{

int a=0, b=1, c, n=10, i;

printf("%d %d",a,b);

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

{

c = a + b ;

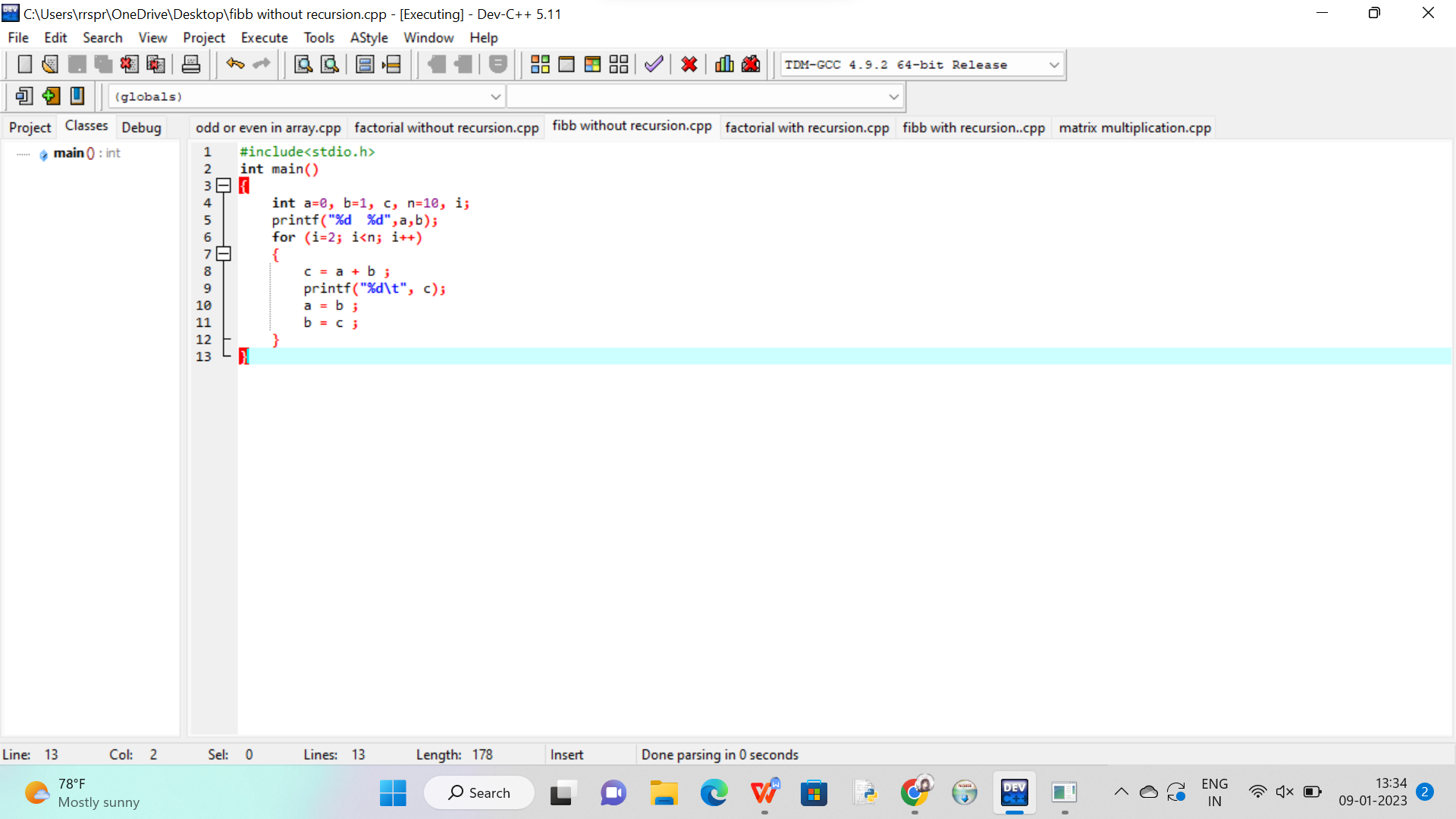
printf("%d\t", c);

a = b ;

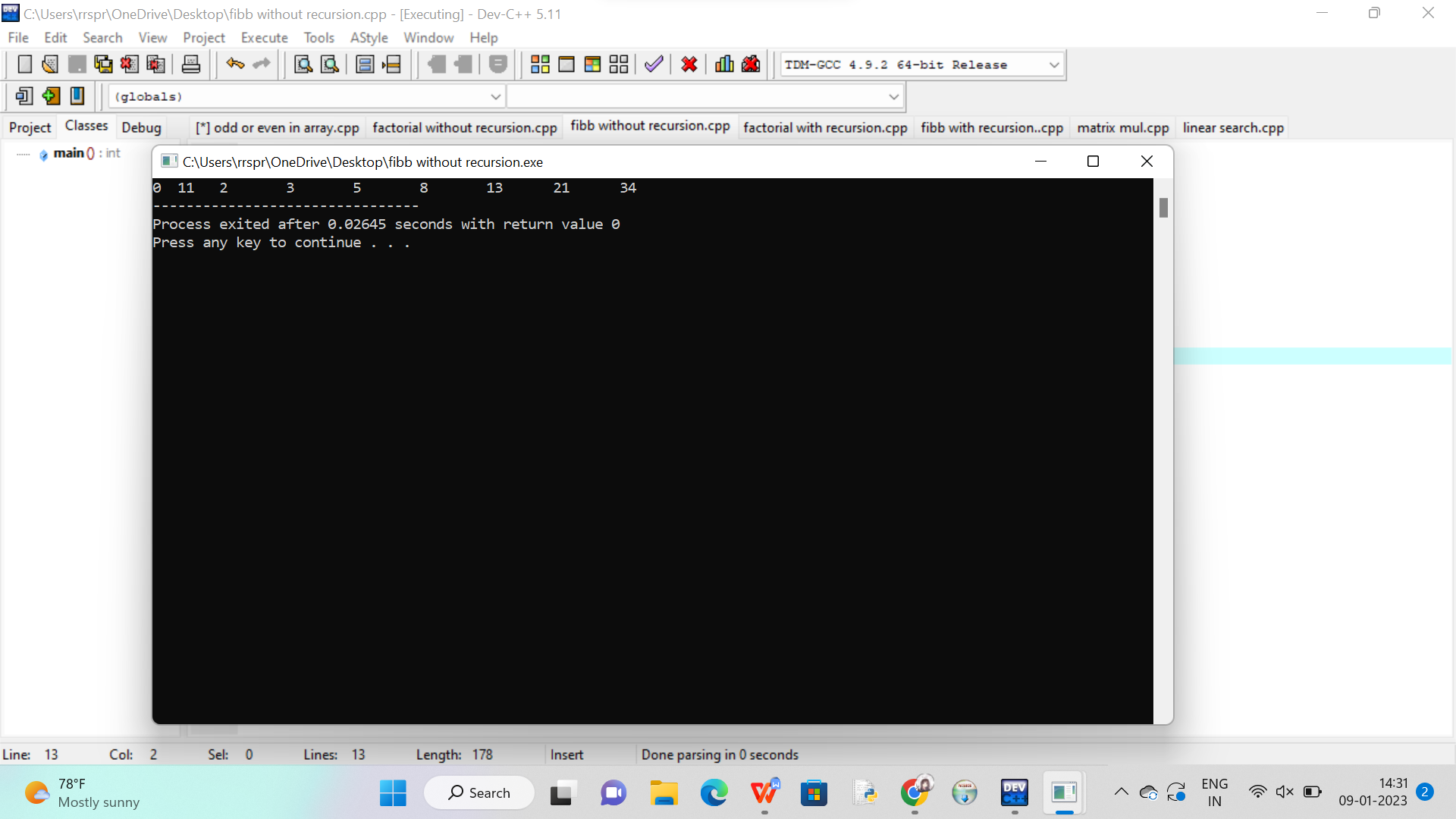
b = c ;

}

}



OUTPUT:



FIBBONACI SERIES WITH RECURSION :

CODE :

#include<stdio.h>

int fibb(int n)

{

if (n==0)

return 0;

else if (n==1)

return 1;

else

return(fibb(n-1)+fibb(n-2));

}

int fibb(int);

int main()

{

int n,i;

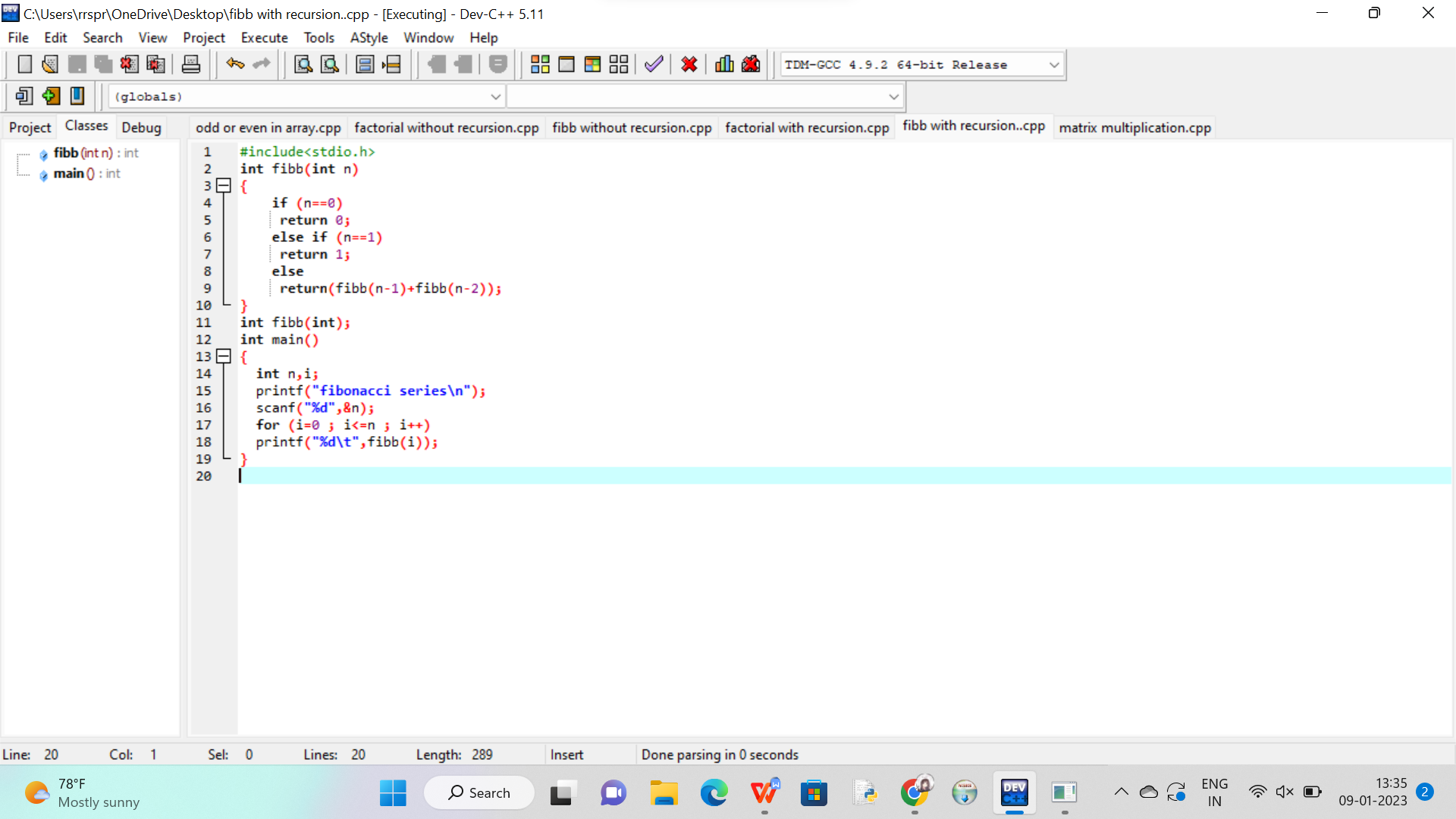
printf("fibonacci series\n");

scanf("%d",&n);

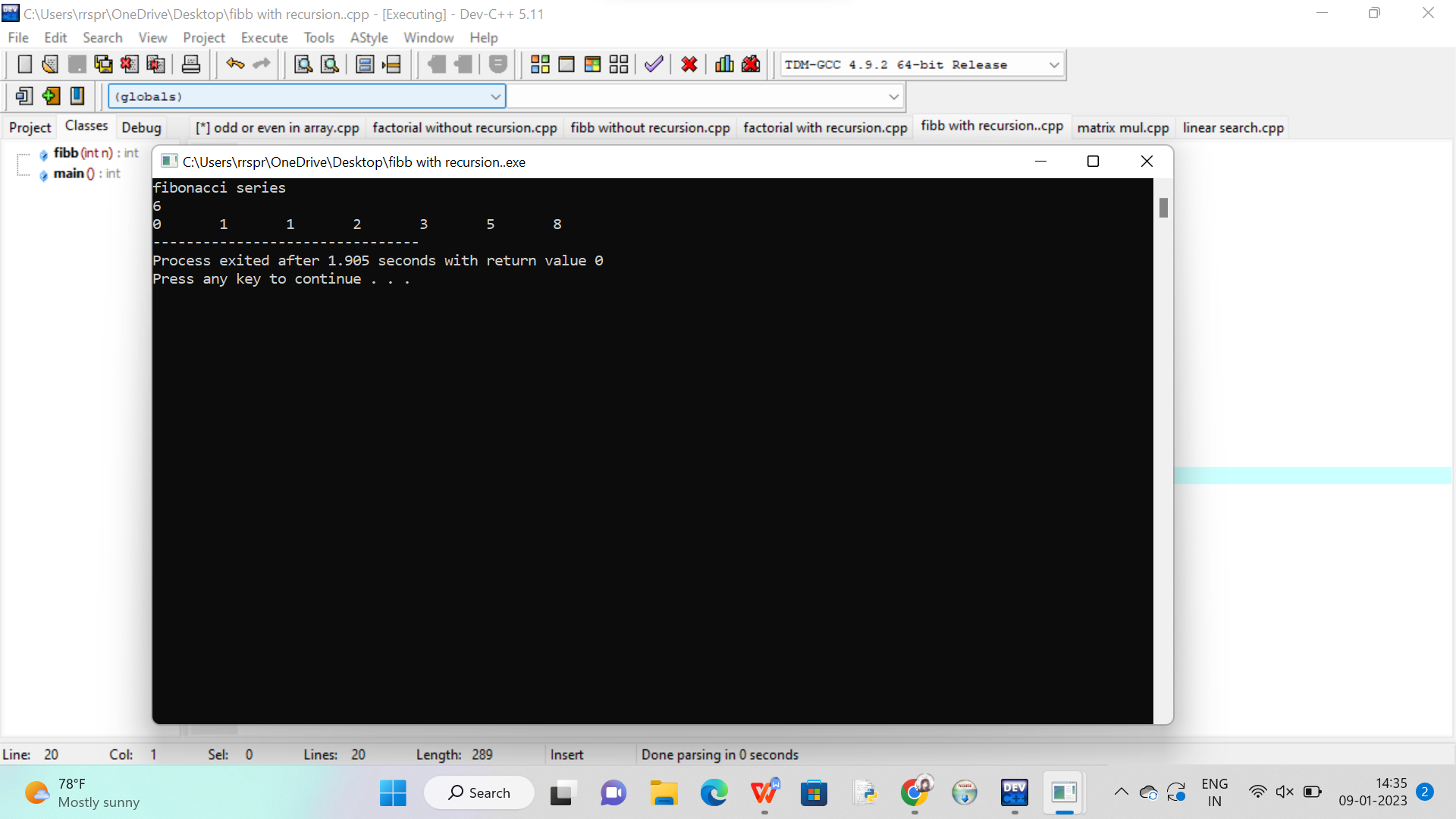
for (i=0 ; i<=n ; i++)

printf("%d\t",fibb(i));

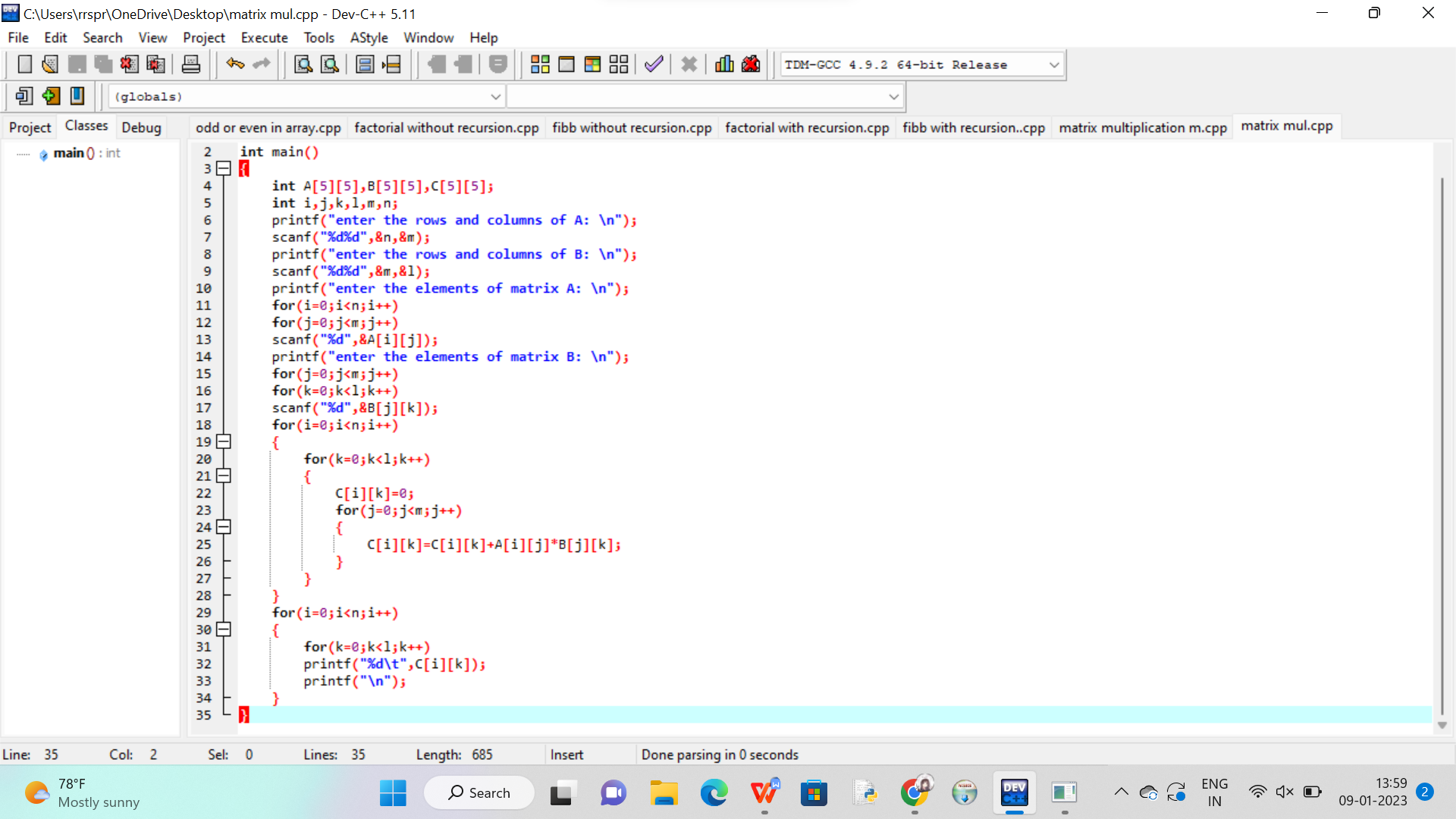
}



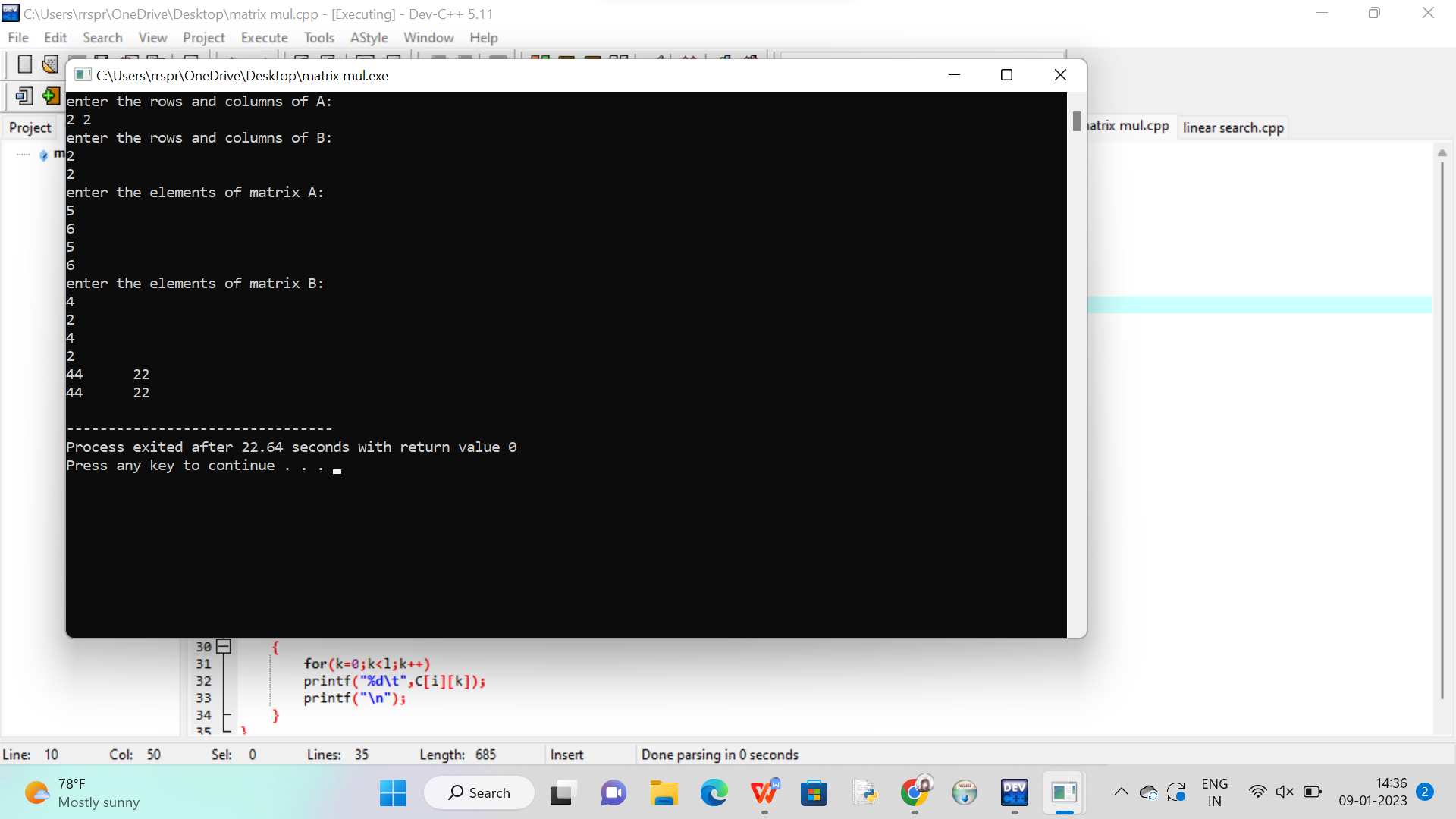
OUTPUT:



Matrix multiplication:



OUT PUT:



int main()

{

int A[5][5],B[5][5],C[5][5];

int i,j,k,l,m,n;

printf("enter the rows and columns of A: \n");

scanf("%d%d",&n,&m);

printf("enter the rows and columns of B: \n");

scanf("%d%d",&m,&l);

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

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

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

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

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

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

for(k=0;k<l;k++)

scanf("%d",&B[j][k]);

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

{

for(k=0;k<l;k++)

{

C[i][k]=0;

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

{

C[i][k]=C[i][k]+A[i][j]\*B[j][k];

}

}

}

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

{

for(k=0;k<l;k++)

printf("%d\t",C[i][k]);

printf("\n");

}

}

Linear search :

Code:

#include<stdio.h>

int main()

{

int a[10],i,n,flag =0,key;

printf("enter the element to be search:\n");

scanf ("%d",&n);

printf("enter the array elements: \n");

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

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

printf("enter the elements to be search:\n");

scanf("%d",&key);

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

{

if (a[i] == key)

{

flag = 1;

break;

}

}

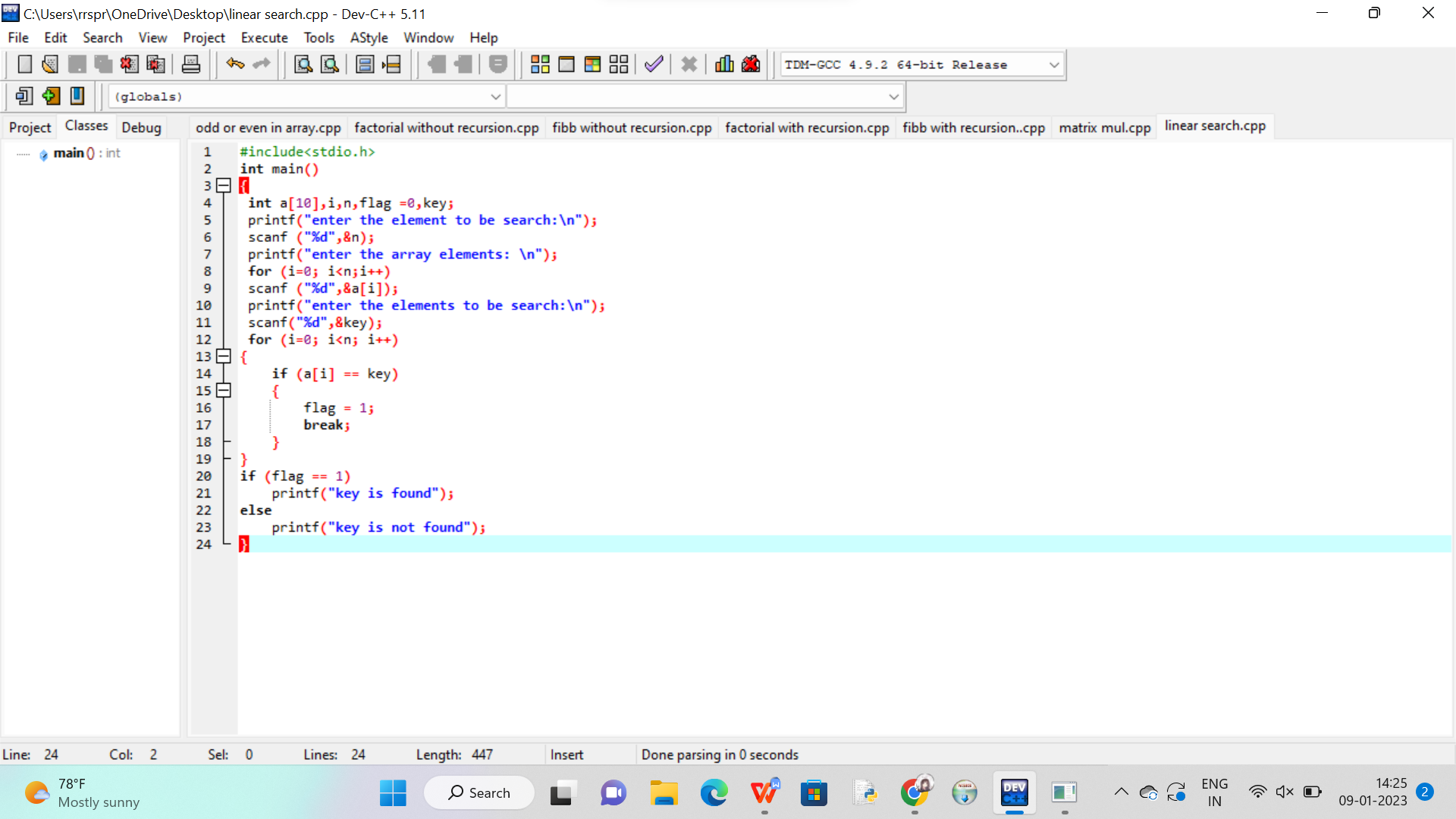
if (flag == 1)

printf("key is found");

else

printf("key is not found");

}



OUT PUT:

