Class-12

Computer all C programming
Practical workouts using function:

1. WAP to print greatest number among two numbers:

PROGRAM:

```
#include<stdio.h>
int greatest(int,int);
int main()
        {
        int a,b;
        printf("enter the numbers: ");
        scanf("%d%d",&a,&b);
        printf("the greatest number is %d",greatest(a,b));
        }
int greatest(int x,int y)
{
        if(x>y)
                {
                   return x;
                 }
        else
                {
                return y;
                 }
}
```

Function prototype:int greatest(int,int);
Return type: int
Variable decleration: a,b;
Variable initialization: --------
Function call: greatest(a,b);

OUTPUT:

```
enter the numbers: 23
 24
the greatest number is 24
```

2. WAP to calculate factorial of given number:

PROGRAM:

```
#include<stdio.h>
int factorial(int);
int main()
{
	int n;
	printf("enter the numbers: ");
	scanf("%d",&n);
	printf("the factorial of given  number is %d",factorial(n));
}
int factorial(int x)
{
	int fact=1,i;
for(i=1;i<=x;i++)
{
	fact=fact*i;
}
return fact;
}
```

Function prototype:factorial(int);
Return type: int
Variable decleration: n,i;
Variable initialization: fact=1;
Function call: factorial(n);

OUTPUT:

```
enter the numbers: 5
the factorial of given  number is 120
```

3. WAP to print multiplication table of n number:

PROGRAM:

```
#include<stdio.h>
void multtable(int);
void main()
{
        int n;
        printf("enter the numbers: ");
        scanf("%d",&n);
        multtable(n);
}
void multtable(int x)
{
        int i;
        for(i=1;i<=10;i++)
        {
    printf("%d*%d=%d\n",x,i,x*i);
    }
}
```

Function prototype:void multtable(int);
Return type: void
Variable decleration: n,i;
Variable initialization: --------
Function call: multtable(n);

OUTPUT:

```
enter the number: 3
3*1=3
3*2=6
3*3=9
3*4=12
3*5=15……..
```

WRITE UP TO 10

4. WAP to display Fibonacci series: 112358….. up to nth term

PROGRAM:

```
#include<stdio.h>
void fibo(int);
void main()
{
        int n;
        printf("enter the number");
        scanf("%d",&n);
        fibo(n);
}
void fibo(int x)
{
        int a=0,b=1,c=1,i;
        for(i=0;i<x;i++)
        {
                printf("%d  ",c);
         c=a+b;
         a=b;
         b=c;
        }
}
```

Function prototype:void fibo(int);
Return type: void
Variable decleration: n,i;
Variable initialization: a=0,b=1,c=1
Function call: fibo(n);

OUTPUT:

```
enter the number4
1  1  2  3
```

5. WAP todisplay the prime series 2 3 5 7…..up to N:

PROGRAM:

```
#include<stdio.h>
int prime(int);
void main()
{
    int x,n,i;
    printf("enter the number up to which you want to print prime
series: ");
    scanf("%d",&n);
    printf("\nthe prime numbers you want are:\n");
    for(i=2;i<=n;i++)
{
        x=prime(i);
       if(x==2)
{
        printf("%d ",i);
}
    }
}
int prime(int x){
    int j,count=0;
    for(j=1;j<=x;j++){
        if(x%j==0){
            count++;
        }
    }
    return count;
}
```

OUTPUT:

```
enter the number up to which you want to print prime series: 5

the prime numbers you want are:
2 3 5
```

↑

PRIME SERIES :

Function prototype:prime(int);
Return type: int
Variable decleration: x,n,i;
Variable initialization: count=0;
Function call: prime(i);

SUM OF ARRAY ELEMENTS :

Function prototype:int sum(int[ ]);
Return type: int
Variable decleration: a[10], i , j ;
Variable initialization: add=0;
Function call: sum(a);

6. WAP to input 5 numbers and calculate their sum using array and function.

PROGRAM:

```
#include<stdio.h>
int sum(int[]);
void main(
){
        int a[10],i;
        printf("enter the elements in array:\n");
        for(i=0;i<5;i++)
{
        scanf("%d",&a[i]);
}
        printf("\nThe sum of given array elements is: %d",sum(a));
}
int sum(int x[])
{
        int j,add=0;
        for(j=0;j<5;j++)
{
                add = add+x[j];
        }
          return add;
}
```

TABLE UPPER PAGE

OUTPUT:

```
enter the elements in array:
1
2
3
4
5

The sum of given array elements is: 15
```

7. WAP to print sum of two numbers using all categories of users defined functions.

a) Function returning values and passing arguments.

PROGRAM:

```
#include<stdio.h>
int sum(int ,int);
void main()
{
	int a,b;
	printf("Enter the numbers: ");
	scanf("%d%d",&a,&b);
	printf("The sum of numbers is %d",sum(a,b));
}
int sum(int x,int z)
{
	return x+z;
}
```

Function prototype:int sum(int,int);
Return type: int
Variable decleration: a,b ;
Variable initialization: ------
Function call: sum(a,b);

OUTPUT:

```
Enter the numbers: 1 2
The sum of numbers is 3
```

b) Function returning no values but passing arguments.

PROGRAM:

```
#include<stdio.h>
void sum(int ,int);
void main()
{
        int a,b;
        printf("Enter the numbers: ");
        scanf("%d%d",&a,&b);
        sum(a,b);
}
void sum(int x,int z)
{
        printf("The sum of numbers is %d",x+z);
}
```

Function prototype:void sum(int,int);
Return type: void
Variable decleration: a,b ;
Variable initialization: ------
Function call: sum(a,b);

OUTPUT:

```
Enter the numbers: 1 2
The sum of numbers is 3
```

c) Function returning values and passing no arguments.

PROGRAM:

```
#include<stdio.h>
int sum();
void main()
{
      printf("The sum of numbers is %d",sum());
}
int sum()
{
   int a,b;
      printf("Enter the numbers: ");
      scanf("%d%d",&a,&b);
      return a+b;
}
```

Function prototype:void sum( );
Return type: int
Variable decleration: a,b ;
Variable initialization: ------
Function call: sum( );

OUTPUT:

```
Enter the numbers: 1 2
The sum of numbers is 3
```

d) Function returning no values and passing no arguments.

PROGRAM:

```
#include<stdio.h>
void sum( );
void main( )
{
   sum( );
}
void sum( )
{
       int a,b;
       printf("Enter the numbers: ");
       scanf("%d%d",&a,&b);
       printf("The sum of numbers is %d",a+b);
}
```

Function prototype:void sum( );
Return type: void
Variable decleration: a,b ;
Variable initialization: ------
Function call: sum( );

OUTPUT:

```
Enter the numbers: 1 2
The sum of numbers is 3
```

8. WAP tocalculate the cumulative sum of n natural numbers using recursion:

PROGRAM:

```
#include<stdio.h>
int cusum(int);
void main(){
	int n;
	printf("Enter the number: ");
	scanf("%d",&n);
	printf("The cumulative sum is %d",cusum(n));
}
int cusum(int x){
	if(x>=1){
		return x+cusum(x-1);
	}
	else{
		return 0;
	}
}
```

Function prototype:int cusum(int);
Return type: int
Variable decleration: n ;
Variable initialization: ------
Function call:cusum(n);

OUTPUT:

```
Enter the number: 5
The cumulative sum is 15
```

10. WAP to count the no of students having weigh between 50 to 60:

PROGRAM:

```
#include<stdio.h>
int count(int[]);
void main(){
	int i,a[10];
	printf("Eenter the weights of students:\n");
	for(i=0;i<5;i++){
		scanf("%d",&a[i]);
	}
	printf("There are %d students having required weight",count(a));
}
int count(int z[]){
	int counter=0,j;
	for(j=0;j<5;j++){
		if(z[j]>50&&z[j]<60){
			counter++;
		}
	}
	return counter;
}
```

Variable initialization: ------

Function call: cumul(n);

OUTPUT:

```
Eenter the weights of students:
12
34
56
54
67
There are 2 students having required weight
```

Program no.10

Function prototype:int count(int [ ]);
Return type: int
Variable decleration: a[10], i , j ;
Variable initialization: counter=0;
Function call: count(a);

Program no.11

Function prototype:int count(int [ ]);
Return type: int
Variable decleration: a[10], i , j ;
Variable initialization: counter=0;
Function call: count(a);

OUTPUT:

```
Enter the size of array you want: 4
Enter the elements of array:
5
4
3
2
The sorted values are:
2 3 4 5
```

11. WAP to sort the given array in ascending order:

PROGRAM:

```
#include<stdio.h>
void sort( int[ ],int);
void main()
{
        int a[100],n,i;
        printf("Enter the size of array you want: ");
        scanf("%d",&n);
                printf("Enter the elements of array: \n");
        for(i=0;i<n;i++)
{
                scanf("%d",&a[i]);
        }
        sort(a,n);
}
void sort(int z[],int x)
{
        int temp,i,j;//sano bata thulo
        for(i=0;i<x-1;i++)
{
                for(j=i+1;j<x;j++)
{
                        if(z[i]>z[j])
{
                                temp=z[i];
                                z[i]=z[j];
                                z[j]=temp;
                        }
                }
        }
printf("The sorted values are:\n");
for(i=0;i<x;i++){
                printf("%d ",z[i]);
        }
}
```

12. WAP to calculate and display multiplication of 3*3 matrix:

PROGRAM:

```
#include<stdio.h>
void matmul(int[10][10],int[10][10]);
int main()
{
int a[10][10],b[10][10],i,j,k;
printf("enter the first matrix element=\n");
for(i=0;i<3;i++)
{
printf("for row %d\n",i);
for(j=0;j<3;j++)
{
scanf("%d",&a[i][j]);
}
printf("\n");
}
printf("enter the second matrix element=\n");
for(i=0;i<3;i++)
{
printf("for row %d\n",i);
for(j=0;j<3;j++)
{
scanf("%d",&b[i][j]);
}
printf("\n");
}
matmul(a,b);
}
```

```
void matmul(int z[10][10],int x[10][10]){
        int i,j,k,mul[10][10];
        printf("PRODUCT OF GIVEN MATRIX IS:\n");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
mul[i][j]=0;
for(k=0;k<3;k++)
{
mul[i][j]+=z[i][k]*x[k][j];
}
}
}
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
printf("%d\t",mul[i][j]);
}
printf("\n");
}
}
```

Function prototype:void matmul(int[10][10],int[10][10]);
Return type: void
Variable decleration: a[10],[10],b[10][10],i,j,k,mul[10][10]
Variable initialization: mul[i][j]=0;
Function call: matmul(a,b);

OUTPUT FOR MATRIX MULTIPLICATION:

```
enter the first matrix element=
for row 0
1
2
3

for row 1
4
5
6

for row 2
7
8
9

enter the second matrix element=
for row 0
1
2
3

for row 1
4
5
6

for row 2
7
8
9

PRODUCT OF GIVEN MATRIX IS:
30      36      42
66      81      96
102     126     150
```

EXTRA QUESTIONS:

1.WAP to calculate and display multiplication of N*N matrix: [USING SINGLE FUNCTION];

PROGRAM:

```
#include<stdio.h>
void input(int[10][10],int[10][10],int ,int ,int ,int);
int main(){
int a[10][10],b[10][10],r1,c1,r2,c2,i,j,k;
printf("enter the number of row and column in first matrix:\n");
scanf("%d%d",&r1,&c1);
printf("enter the number of row and column in second
matrix:\n");
scanf("%d%d",&r2,&c2);
if(c1==r2){
printf("enter the first matrix element=\n");
for(i=0;i<r1;i++)
{
printf("for row %d\n",i);
for(j=0;j<c1;j++)  {
scanf("%d",&a[i][j]);
}
printf("\n");
}
printf("enter the second matrix element=\n");
for(i=0;i<r2;i++) {
printf("for row %d\n",i);
for(j=0;j<c2;j++){
scanf("%d",&b[i][j]);
}
printf("\n");
}
input(a,b,r1,c1,r2,c2);
}
else{
        printf("!!!INVALID ROWS AND COLUMN!!!");
}
}
```

```
void input(int z[10][10],int x[10][10],int r1,int c1,int r2,int c2){
	int i,j,k,mul[10][10];
	printf("PRODUCT OF GIVEN MATRIX IS:\n");
for(i=0;i<r1;i++)
{
for(j=0;j<c2;j++)
{
mul[i][j]=0;
for(k=0;k<c1;k++)
{
mul[i][j]+=z[i][k]*x[k][j];
}
}
}
for(i=0;i<r1;i++)
{
for(j=0;j<c2;j++)
{
printf("%d\t",mul[i][j]);
}
printf("\n");
}
}
```

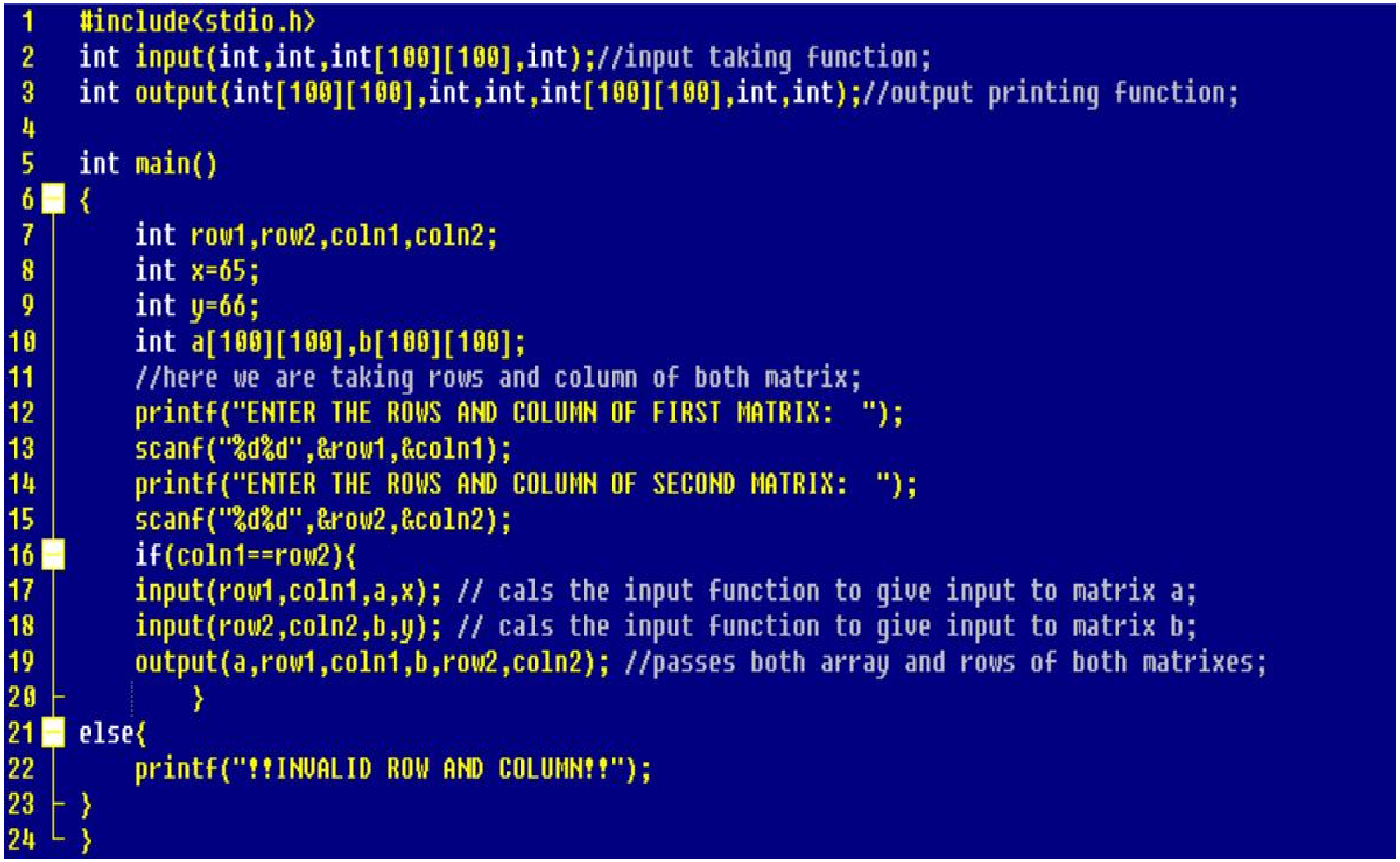
YOU CAN COMPILE AND SEE THE RESULT:::

1.WAP to calculate and display multiplication of N*N matrix: [USING MULTIPLE FUNCTION];

PROGRAM:

NOTE: FOR THIS COMPLEX CODE I HAVE KEPT THE PICTURE SO THAT IT WILL BE A BIT EASIER:

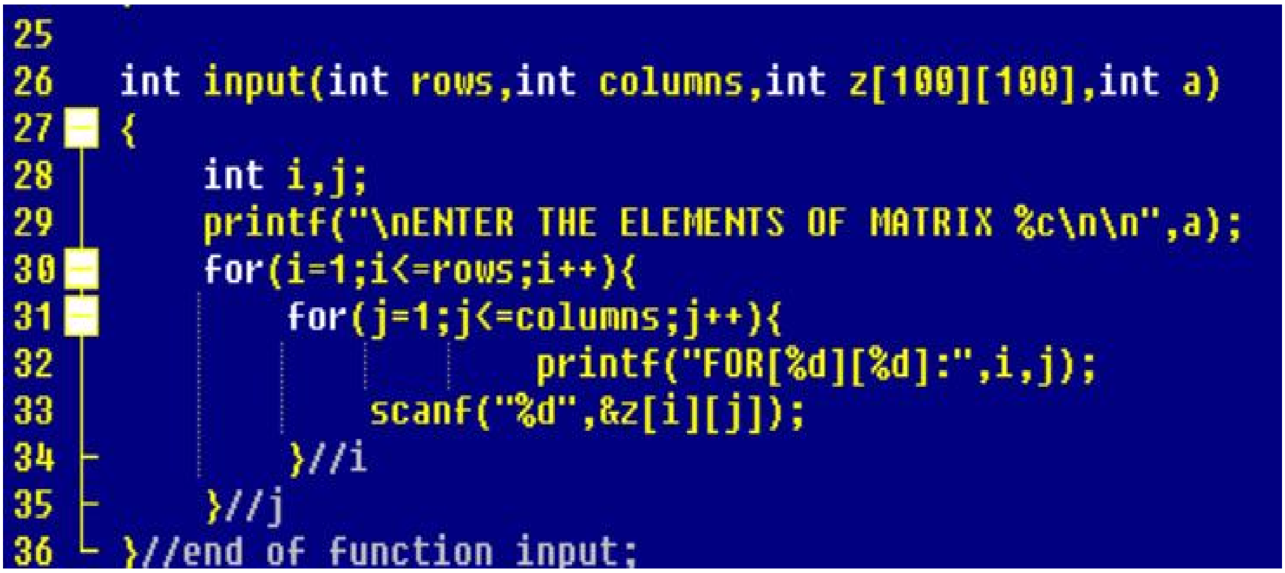
MAIN FUNCTION:



```
#include<stdio.h>
int input(int,int,int[100][100],int);//input taking function;
int output(int[100][100],int,int,int[100][100],int,int);//output printing function;

int main()
{
    int row1,row2,coln1,coln2;
    int x=65;
    int y=66;
    int a[100][100],b[100][100];
    //here we are taking rows and column of both matrix;
    printf("ENTER THE ROWS AND COLUMN OF FIRST MATRIX:  ");
    scanf("%d%d",&row1,&coln1);
    printf("ENTER THE ROWS AND COLUMN OF SECOND MATRIX:  ");
    scanf("%d%d",&row2,&coln2);
    if(coln1==row2){
    input(row1,coln1,a,x); // cals the input function to give input to matrix a;
    input(row2,coln2,b,y); // cals the input function to give input to matrix b;
    output(a,row1,coln1,b,row2,coln2); //passes both array and rows of both matrixes;
        }
else{
    printf("!!INVALID ROW AND COLUMN!!");
}
}
```

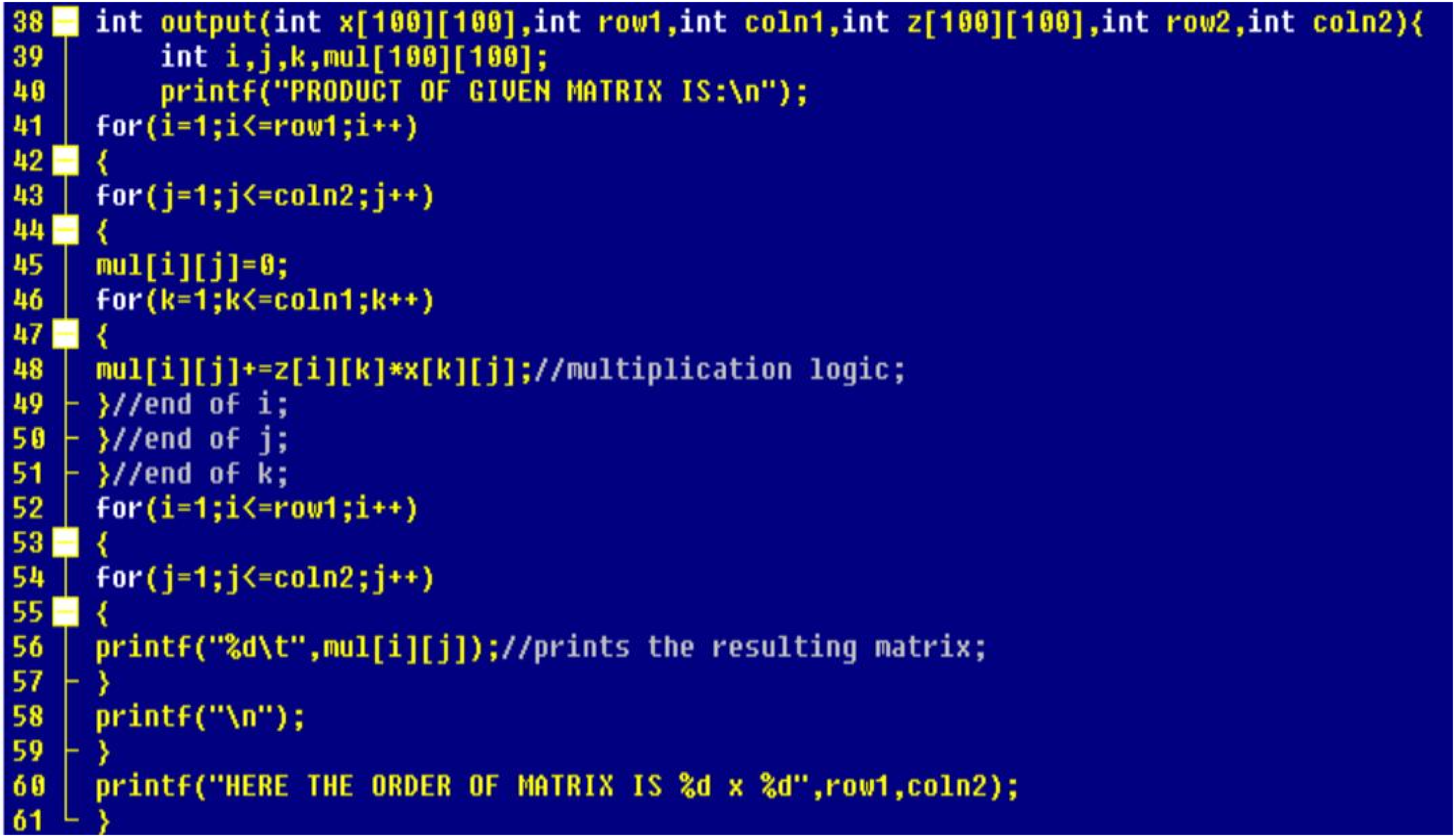
INPUT TAKING FUNCTION:



```

int input(int rows,int columns,int z[100][100],int a)
{
    int i,j;
    printf("\nENTER THE ELEMENTS OF MATRIX %c\n\n",a);
    for(i=1;i<=rows;i++){
        for(j=1;j<=columns;j++){
                    printf("FOR[%d][%d]:",i,j);
            scanf("%d",&z[i][j]);
        }//i
    }//j
}//end of function input;
```

OUTPUT GIVING FUNCTION:



```
int output(int x[100][100],int row1,int coln1,int z[100][100],int row2,int coln2){
    int i,j,k,mul[100][100];
    printf("PRODUCT OF GIVEN MATRIX IS:\n");
for(i=1;i<=row1;i++)
{
for(j=1;j<=coln2;j++)
{
mul[i][j]=0;
for(k=1;k<=coln1;k++)
{
mul[i][j]+=z[i][k]*x[k][j];//multiplication logic;
}//end of i;
}//end of j;
}//end of k;
for(i=1;i<=row1;i++)
{
for(j=1;j<=coln2;j++)
{
printf("%d\t",mul[i][j]);//prints the resulting matrix;
}
printf("\n");
}
printf("HERE THE ORDER OF MATRIX IS %d x %d",row1,coln2);
}
```

Don’t worry here is the code:

MAIN FUNCTION:

```
#include<stdio.h>
int input(int,int,int[100][100],int);//input taking function;
int output(int[100][100],int,int,int[100][100],int,int);//output printing
function;

int main()
{
        int row1,row2,coln1,coln2;
        int x=65;
        int y=66;
        int a[100][100],b[100][100];
        //here we are taking rows and column of both matrix;
        printf("ENTER THE ROWS AND COLUMN OF FIRST
MATRIX:  ");
    scanf("%d%d",&row1,&coln1);
        printf("ENTER THE ROWS AND COLUMN OF SECOND
MATRIX:  ");
        scanf("%d%d",&row2,&coln2);
        if(coln1==row2){
        input(row1,coln1,a,x); // cals the input function to give input to
matrix a;
        input(row2,coln2,b,y); // cals the input function to give input to
matrix b;
        output(a,row1,coln1,b,row2,coln2); //passes both array and rows
of both matrixes;
                }
else{
        printf("!!INVALID ROW AND COLUMN!!");
}
}
```

INPUT FUNCTION:

```
int input(int rows,int columns,int z[100][100],int a)
{
    int i,j;
    printf("\nENTER THE ELEMENTS OF MATRIX %c\n\n",a);
    for(i=1;i<=rows;i++){
        for(j=1;j<=columns;j++){
                    printf("FOR[%d][%d]:",i,j);
            scanf("%d",&z[i][j]);
        }//i
    }//j
}//end of function input;
```

OUTPUT FUNCTION:

```
int output(int x[100][100],int row1,int coln1,int z[100][100],int
row2,int coln2){
    int i,j,k,mul[100][100];
    printf("PRODUCT OF GIVEN MATRIX IS:\n");
for(i=1;i<=row1;i++)
{
for(j=1;j<=coln2;j++)
{
mul[i][j]=0;
for(k=1;k<=coln1;k++)
{
mul[i][j]+=z[i][k]*x[k][j];//multiplication logic;
}//end of i;
}//end of j;
}//end of k;
for(i=1;i<=row1;i++)
{
for(j=1;j<=coln2;j++)
{
printf("%d\t",mul[i][j]);//prints the resulting matrix;
}
printf("\n");
}
printf("HERE THE ORDER OF MATRIX IS %d x %d",row1,coln2);
}
```

If you have any quires then you can inbox me and also you guys can ask me for codes.

:::::THANK-YOU:::::