

## #include void enterData(int

```
#include <stdio.h> void enterData(int firstMatrix[][10], int secondMatrix[][10], int rowFirst, int
columnFirst, int rowSecond, int columnSecond); void multiplyMatrices(int firstMatrix[][10], int
secondMatrix[][10], int multResult[][10], int rowFirst, int columnFirst, int rowSecond, int
columnSecond); void display(int mult[][10], int rowFirst, int columnSecond); int main() { int
firstMatrix[10][10], secondMatrix[10][10], mult[10][10], rowFirst, columnFirst, rowSecond,
columnSecond, i, j, k; printf("Enter rows and column for first matrix: "); scanf("%d %d", &rowFirst,
&columnFirst); printf("Enter rows and column for second matrix: "); scanf("%d %d", &rowSecond,
&columnSecond); // If column of first matrix is not equal to row of second matrix, asking user to enter
the size of matrix again. while (columnFirst != rowSecond) { printf("Error! column of first matrix not
equal to row of second.\n"); printf("Enter rows and column for first matrix: "); scanf("%d %d", &rowFirst,
&columnFirst); printf("Enter rows and column for second matrix: "); scanf("%d %d", &rowSecond,
&columnSecond); } // Function to take matrices data enterData(firstMatrix, secondMatrix, rowFirst,
columnFirst, rowSecond, columnSecond); // Function to multiply two matrices.
multiplyMatrices(firstMatrix, secondMatrix, mult, rowFirst, columnFirst, rowSecond, columnSecond); //
Function to display resultant matrix after multiplication. display(mult, rowFirst, columnSecond); return
0; } void enterData(int firstMatrix[][10], int secondMatrix[][10], int rowFirst, int columnFirst, int
rowSecond, int columnSecond) { int i, j; printf("\nEnter elements of matrix 1:\n"); for(i = 0; i < rowFirst;
++i) { for(j = 0; j < columnFirst; ++j) { printf("Enter elements a%d%d: ", i + 1, j + 1); scanf("%d",
&firstMatrix[i][j]); } } printf("\nEnter elements of matrix 2:\n"); for(i = 0; i < rowSecond; ++i) { for(j = 0; j <
columnSecond; ++j) { printf("Enter elements b%d%d: ", i + 1, j + 1); scanf("%d", &secondMatrix[i][j]); } }
} void multiplyMatrices(int firstMatrix[][10], int secondMatrix[][10], int mult[][10], int rowFirst, int
columnFirst, int rowSecond, int columnSecond) { int i, j, k; // Initializing elements of matrix mult to 0.
for(i = 0; i < rowFirst; ++i) { for(j = 0; j < columnSecond; ++j) { mult[i][j] = 0; } } // Multiplying matrix
firstMatrix and secondMatrix and storing in array mult. for(i = 0; i < rowFirst; ++i) { for(j = 0; j <
columnSecond; ++j) { for(k=0; k<columnFirst; ++k) { mult[i][j] += firstMatrix[i][k] * secondMatrix[k][j]; } }
} void display(int mult[][10], int rowFirst, int columnSecond) { int i, j; printf("\nOutput Matrix:\n"); for(i =
0; i < rowFirst; ++i) { for(j = 0; j < columnSecond; ++j) { printf("%d ", mult[i][j]); if(j == columnSecond - 1)
printf("\n\n"); } } }
```

### Output

Enter rows and column for first matrix: 3 2 Enter rows and column for second matrix: 3 2 Error! column of first matrix not equal to row of second. Enter rows and column for first matrix: 2 3 Enter rows and column for second matrix: 3 2 Enter elements of matrix 1: Enter elements a11: 3 Enter elements a12: -2 Enter elements a13: 5 Enter elements a21: 3 Enter elements a22: 0 Enter elements a23: 4 Enter elements of matrix 2: Enter elements b11: 2 Enter elements b12: 3 Enter elements b21: -9 Enter elements b22: 0 Enter elements b31: 0 Enter elements b32: 4 Output Matrix: [24 29 6 25](#)

```

d) #include <stdio.h>
int fact (int) ;
void main ( )
{
    int no, factorial ;
    printf ( "Enter a number to calculate
             it's factorial \n" );
    scanf ( "%d", &no );
    factorial = fact (no) ;
    printf ( "factorial of the number (%d)
             = %d \n", no, factorial );
}
int fact ( int n )
{
    int i, f = 1 ;
    for ( i = 1 ; i <= n ; i++ )
    {
        f = f * i ;
    }
    return f ;
}

```

Output :

Enter a number to calculate it's factorial  
5.

factorial of the num (5) = 120.

```

#include <stdio.h>
#include <conio.h>
void main ()
{
    int num, res = 0;
    clrscr();
    printf ( "\n Enter a number : " );
    scanf ( "%d", &num);
    res = prime (num);
    if ( res == 0 )
        printf ( "\n %d is a prime number", num );
    else
        printf ( "\n %d is not a prime number",
                num );
    getch ();
}

int prime ( int n )
{
    int i;
    for ( i = 2; i <= n / 2; i++ )
    {
        if ( n % i == 0 )
            continue;
        else
            return 0;
    }
    return 1;
}

```

Output : Enter a number : 7  
7 is a prime number.