ARRAYS

<Arrays – Initialization – Declaration – One dimensional and Two-dimensional arrays.>

1. ONE DIMENTIONAL ARRAY

2. TWO DIMENTIONAL ARRAY

3. STRING ARRAYS (ONE DIMENTIONAL ARRAY and TWO DIMENTIONAL ARRAY)

4. MULTIDIMENTIONAL ARRAYS

An **array** is a collection of similar data items, accessed using a common name. The collection of element can all be integers or be all decimal value or be all characters or be all strings.

- A one-dimensional **array** is like a list
- A two dimensional **array** is like a table
- The **C** language places no limits on the number of dimensions in an **array**

ONE DIMENTIONAL ARRAY

Array Declaration:

To declare an array in C, a programmer specifies the type of the elements and the number of elements required. The **arraySize** must be an integer constant greater than zero and **datatype** can be any valid C data type.

Syntax1:

```
datatype arrayName[ arraySize ];
```

Example-1

```
int number[20];
int marks[44];
float salary[10];
double value[25];
```

```
int n=25;
double x[n], y[n]; //array
                   declaration
```

```
int  n;
Scanf("%d",&n);//get size
int x[n]; //array declaration
```

```
#include<stdio.h>
#define N 100
int main( )
{
int marks[N];//array dec
....
return 0;
}
```

```
#include<stdio.h>
int main( )
{
int N=10,M=20;
int marks[N*M];//array dec
....
return 0;
}
```

Syntax-2

```
<storage class> datatype arrayName[ arraySize ];
```

Example: static int marks[20];

Array intitialization:

Example-1 `int mark[5] = {55, 66, 77, 88, 99};`

```
mark[0] = 55
mark[1] = 66
mark[2] = 77
mark[3] = 88
mark[4] = 99
```

	mark[0]	mark[1]	mark[2]	mark[3]	mark[4]
array elements value	55	66	77	88	99
array index	0	1	2	3	4

Example-2 `double balance[] = {1000.0, 2.0, 3.4, 7.0, 50.0};`

it means....

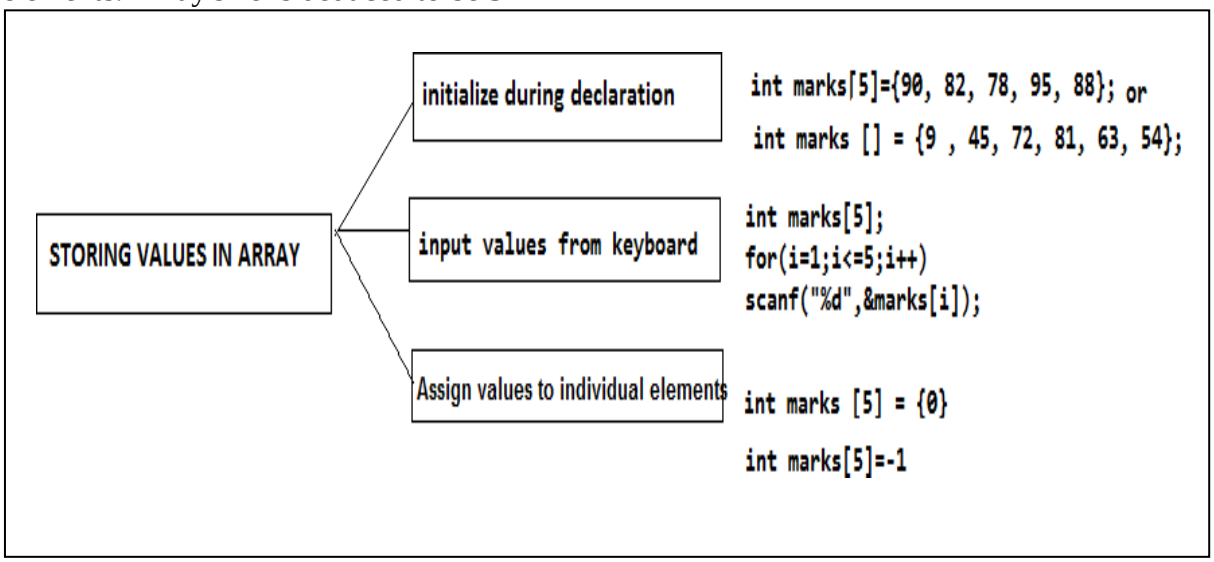
```
balance[0] = 1000.0;
balance[1] = 2.0;
balance[2] = 3.4;
balance[3] = 7.0;
balance[4] = 50.0;
```

	0	1	2	3	4
balance	1000.0	2.0	3.4	7.0	50.0

Automatic sizing

int arr[] = {3,1,5,7,9};

Here, the C compiler will deduce the size of the array automatically based on the number of elements. Array size is deduced to be 5



OPERATIONS ON ARRAYS

- **Traversing an array**
- **Inserting an element in an array**
- **Searching an element in an array**
- **Deleting an element from an array**
- **Merging two arrays**
- **Sorting an array in ascending or descending order**

Working with one dimensional array

STORE and DISPLAY VALUES IN AN ARRAY (traversing an array)

```
#include<stdio.h>
int main( )
{
int k,array[10];//array declaration
printf("Enter the array elements:");
for(k=0;k<5;k++)
{
scanf("%d ",&array[i]); // storing values in array
}
printf("\n Display the array elements:");
for(k=0;k<5;k++)
{
printf("%d \n",array[i]);//displaying values of array
}
return 0;
}
```

Output:

```
Enter the array elements
2
4
3
1
8
Display the array elements
2
4
3
1
8
```

FIND SUM AND AVERAGE OF N NUMBERS

```
#include<stdio.h>
int main( )
{
int k,n,sum=0;array[10];//array declaration
float avg;
printf("\n Enter the array size:");
scanf("%d",&n);
printf("\n Enter the array elements:");
for(k=0;k<n;k++)
{
scanf("%d ",&array[i]); // storing values in array
}
for(k=0;k<n;k++)
{
sum=sum+array[i]; //sum of array elements
}
avg=sum/n;
printf("\n sum=%d and  avg=%f  ",sum,avg);
}
return 0;
}
```

Output:

```
Enter the array size: 6
Enter the array elements
9
2
4
3
1
8

sum=27 and avg=4.50000
```

REVERSE OF ARRAY ELEMENTS

```
#include<stdio.h>
int main( )
{
int k,n,array[10];//array declaration
printf("\n Enter the array size:");
scanf("%d",&n);
printf("\n Enter the array elements:");
for(k=0;k<n;k++)
```

Output:

```
Enter the array elements
2
4
3
1
8
Display the array elements
8
1
3
4
2
```

```
{
scanf("%d ",&array[i]); // storing values in array
}
printf("\n array elements in reverse order:");
for(k=n-1;k>=0;k--)
{
printf("%d \n",array[i]);//displaying values of array
}
return 0;
}
```

Write a program to print the position of the smallest number of *n* numbers using arrays.

```
#include <stdio.h>
int main()
{
int i, n, arr[20], small, pos;
printf("\n Enter the number of elements in the array : ");
scanf("%d", &n);
printf("\n Enter the elements : ");
for(i=0;i<n;i++)
scanf("%d",&arr[i]);
small = arr[0]
for(i=1;i<n;i++)
{
if(arr[i]<small)
{
small = arr[i];
pos = i;
}
}
printf("\n The smallest element is : %d", small);
printf("\n The position of the smallest element in the array is:
%d", pos);
return 0;
}
```

Output

```
Enter the number of elements in the array : 5
Enter the elements : 7 6 5 14 3
The smallest element is : 3
The position of the smallest element in the
array is : 4
```

Program example-1 Printing binary equivalent of a decimal number using array

Logic→ Here the remainders of the integer division of a decimal number by 2 are stored as consecutive array elements.The division procedure is repeated until the number becomes 0.

```
#include <stdio.h>
int main()
{
int  bi[20],i,m,num,rem;
printf("\n Enter the decimal Integer");
scanf("%d",&n);
m=n;
for(i=0;i>n;i++)
{
rem=num%2;
```

Output:

```
Enter the decimal Integer: 12
Binary equivalent of 12 is: 1100
```

```
bi[i]=rem;
num=num/2;
}
printf("\n Binary equivalent of %d is: \t",m);
for(i--;i>=0;i--)
printf("%d",a[i]);
return 0;
}
```

Program example- Fibonacci series using an array

Logic→ In **Fibonacci series** each element is the sum of the previous two elements. This program stores the series in an array

```
#include <stdio.h>
int main()
{
int fib[15]; // array declaration
int i;
fib[0] = 0; // first array element value=0
fi b[1] = 1;// second array element value=1
for(i = 2; i < 15; i++)
{
 fib[i] = fib[i-1] + fib[i-2];
}
printf("\n Display the fibonacci elements:");
for(i = 0; i < 15; i++)
{
printf("%d\n", fi b[i]);
}
return 0;
}
```

Display the fibonacci elements

0
1
1
2
3
5
8
13
21
34
55
89
144
233
377

Example- Inserting an Element in an Array

If an element has to be inserted at the end of an existing array, then the task of insertion is quite simple. We just have to add 1 to the upper_ bound and assign the value. Here, we assume that the memory space allocated for the array is still available. For example, if an array is declared to contain 10 elements, but currently it has only 8 elements, then obviously there is space to accommodate two more elements. But if it already has 10 elements, then we will not be ableto add another element to it.

Program to insert a number at a given location in an array

```
#include <stdio.h>
int main()
{
int i, n, num, pos, arr[10];
clrscr();
printf("\n Enter the number of elements in the array : ");
scanf("%d", &n);
for(i=0;i<n;i++)
{
printf("\n arr[%d] = ", i);
scanf("%d", &arr[i]);
}
printf("\n Enter the number to be inserted : ");
scanf("%d", &num);
printf("\n Enter the position at which the number has to be added: ");
```

```
for(i=n-1;i>=pos;i--)
arr[i+1] = arr[i];
arr[pos] = num;
n = n+1;
printf("\n The array after insertion of %d is :num ");
for(i=0;i<n;i++)
printf("\n arr[%d] = %d", i, arr[i]);
getch();
return 0;
}
```

Output

```
Enter the number of elements in the array : 5
arr[0] = 1
arr[1] = 2
arr[2] = 3
arr[3] = 4
arr[4] = 5
Enter the number to be inserted : 0
Enter the position at which the number has to be added : 3
The array after insertion of 0 is :
arr[0] = 1
arr[1] = 2
arr[2] = 3
arr[3] = 0
arr[4] = 4
arr[5] = 5
```

3.Deleting an Element from an Array

Algorithm to delete an element from the middle of an array

Step 1: [INITIALIZATION] SET I = POS
Step 2: Repeat Steps 3 and 4 while I <= N – 1
Step 3: SET A[I] = A[I + 1]
Step 4: SET I = I + 1
[END OF LOOP]
Step 5: SET N = N – 1
Step 6: EXIT

Write a program to delete a number from a given location in an array.

```
#include <stdio.h>
int main()
{
int i, n, pos, arr[10];
printf("\n Enter the number of elements in the array : ");
scanf("%d", &n);
for(i=0;i<n;i++)
{
printf("\n arr[%d] = ", i);
scanf("%d", &arr[i]);
}
printf("\nEnter the position from which the number has to be deleted : ");
```

```
for(i=pos; i<n–1;i++)
arr[i] = arr[i+1];
n– –;
printf("\n The array after deletion is : ");
for(i=0;i<n;i++)
printf("\n arr[%d] = %d", i, arr[i]);
getch();
return 0;
}
```

Output

```
Enter the number of elements in the array :
5
arr[0] = 1
arr[1] = 2
arr[2] = 3
arr[3] = 4
arr[4] = 5
Enter the position from which the number
has to be deleted : 3
The array after deletion is :
arr[0] = 1
arr[1] = 2
arr[2] = 3
arr[3] = 5
```

TWO DIMENTIONAL ARRAY

Two dimentional arrays stores data in tabular column format represented as rows and columns

Array Declaration:

datatype arrayname[size][size];

Array Initialization:

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

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

1 4
2 3

float[ ][ ]={12.3, 45.2,19.3,23.4}

12.3 45.2

19.3 23.4

Accessing two-dimensional Arrays

Program-sample two dimnentional array

```
#include <stdio.h>
int main()
{
int i,j;
int a[3][2] = {{4,7},{1,0},{6,2}};
for(i = 0; i < 3; i++)
{
for(j = 0; j < 2; j++)
{
printf("%d", a[i][j]);
}
printf("\n");
}
return 0;
}
```

Row-1	**4**	**7**
Row -2	**1**	**0**
Row-3	**6**	**2**

The above array actually 'looks' like this

1	2	3	4	5	6	7	8	9
Row 0			Row 1			Row 2		

WORKING WITH TWO-DIMENSIONAL ARRAYS

Transpose of a matrix

Example program:-Transpose of a matrix

Transpose of *A* is *AT*=(*aji*), where *i* is the row number and *j* is the column number.

$$A=\begin{pmatrix}5 & 2 & 3\\ 4 & 7 & 1\\ 8 & 9 & 9\end{pmatrix} \qquad A^T=\begin{pmatrix}5 & 4 & 8\\ 2 & 7 & 9\\ 3 & 1 & 9\end{pmatrix}$$

Program

```
#include <stdio.h>

int main()
{
   int a[10][10], transpose[10][10], r, c, i, j;
   printf("Enter rows and columns of matrix: ");
   scanf("%d %d", &r, &c);

   // getting elements of the matrix
   printf("\nEnter elements of matrix:\n");
   for(i=0; i<r; ++i)
      for(j=0; j<c; ++j)
      {
         printf("Enter element a%d%d: ",i+1, j+1);
         scanf("%d", &a[i][j]);
      }

   // Displaying the matrix a[][] */
   printf("\n Entered Matrix: \n");
   for(i=0; i<r; ++i)
      for(j=0; j<c; ++j)
      {
         printf("%d  ", a[i][j]);
         if (j == c-1)
            printf("\n\n");
      }

   // Finding the transpose of matrix a
   for(i=0; i<r; ++i)
      for(j=0; j<c; ++j)
      {
         transpose[j][i] = a[i][j];
      }

   // Displaying the transpose of matrix a
   printf("\nTranspose of Matrix:\n");
   for(i=0; i<c; ++i)
      for(j=0; j<r; ++j)
      {
         printf("%d  ",transpose[i][j]);
         if(j==r-1)
            printf("\n\n");
      }

   return 0;
}
```

Sample output

```
Enter rows and columns of matrix: 2
3

Enter element of matrix:
Enter element a11: 2
Enter element a12: 3
Enter element a13: 4
Enter element a21: 5
Enter element a22: 6
Enter element a23: 4

Entered Matrix:
2 3  4

5  6  4


Transpose of Matrix:
2     5

3     6

4     4
```

Program -2(Transpose)

```
#include <stdio.h>

void main()
{
    int array[10][10];
  int i, j, m, n;

   printf("Enter the order of the matrix \n");
 scanf("%d %d", &m, &n);
 printf("Enter the coefiicients of the matrix\n");
 for (i = 0; i < m; ++i)
{
        for (j = 0; j < n; ++j)
    {
            scanf("%d", &array[i][j]);
        }
    }
    printf("The given matrix is \n");
    for (i = 0; i < m; ++i)
    {
        for (j = 0; j < n; ++j)
        {
            printf(" %d", array[i][j]);
        }
        printf("\n");
    }
    printf("Transpose of matrix is \n");
    for (j = 0; j < n; ++j)
    {
        for (i = 0; i < m; ++i)
        {
            printf(" %d", array[i][j]);
        }
        printf("\n");
    }
```

```
$ cc pgm85.c
$ a.out
Enter the order of the matrix
3 3
Enter the coefiicients of the matrix
3 7 9
2 7 5
6 3 4
The given matrix is
 3 7 9
 2 7 5
 6 3 4
Transpose of matrix is
 3 2 6
 7 7 3
 9 5 4
```

Matrix addition and subtraction

Addition If *A* and *B* above are matrices of the same type

$$A+B=\begin{pmatrix}1 & 2 & 3\\ 1 & 0 & 2\end{pmatrix}+\begin{pmatrix}2 & 1 & 2\\ 1 & 0 & 3\end{pmatrix}=\begin{pmatrix}3 & 3 & 5\\ 2 & 0 & 5\end{pmatrix}$$

Subtraction If *A* and *B* are matrices of the same type, then

$$A-B=\begin{pmatrix}1 & 2 & 3\\ 1 & 0 & 2\end{pmatrix}-\begin{pmatrix}2 & 1 & 2\\ 1 & 0 & 3\end{pmatrix}=\begin{pmatrix}-1 & 1 & 1\\ 0 & 0 & -1\end{pmatrix}$$

Program to Add Two Matrices

```
#include <stdio.h>
int main(){
   int r, c, a[100][100], b[100][100], sum[100][100], i, j;

   printf("Enter number of rows (between 1 and 100): ");
   scanf("%d", &r);
   printf("Enter number of columns (between 1 and 100): ");
   scanf("%d", &c);
   printf("\nEnter elements of 1st matrix:\n");
   for(i=0; i<r; ++i)
      for(j=0; j<c; ++j)
      {
         printf("Enter element a%d%d: ",i+1,j+1);
         scanf("%d",&a[i][j]);
      }
   printf("Enter elements of 2nd matrix:\n");
   for(i=0; i<r; ++i)
      for(j=0; j<c; ++j)
      {
         printf("Enter element a%d%d: ",i+1, j+1);
         scanf("%d", &b[i][j]);
      }
   // Adding Two matrices
   for(i=0;i<r;++i)
      for(j=0;j<c;++j)
      {
         sum[i][j]=a[i][j]+b[i][j];
      }
  // Displaying the result
   printf("\nSum of two matrix is: \n\n");
   for(i=0;i<r;++i)
      for(j=0;j<c;++j)
      {
         printf("%d  ",sum[i][j]);
         if(j==c-1)
         {
            printf("\n\n");
         }
      }
      return 0;
}
```

Output

```
Enter number of rows (between
1 and 100): 2
Enter number of columns
(between 1 and 100): 3

Enter elements of 1st matrix:
Enter element a11: 2
Enter element a12: 3
Enter element a13: 4
Enter element a21: 5
Enter element a22: 2
Enter element a23: 3
Enter elements of 2nd matrix:
Enter element a11: -4
Enter element a12: 5
Enter element a13: 3
Enter element a21: 5
Enter element a22: 6
Enter element a23: 3

Sum of two matrix is:

-2   8   7

10   8   6
```

Matrix multiplication

Matrix multiplication for two 2 × 2 matrices.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} e & f \\ g & h \end{pmatrix}=\begin{pmatrix} (ae+bg) & (af+bh) \\ (ce+dg) & (ef+dh) \end{pmatrix}$$

Finding norm of a matrix

The norm of a matrix is defi ned as the square root of the sum of the squares of the elements of a matrix.

```
#include <stdio.h>
#include <math.h>
#defi ne row 10
#defi ne col 10
int main()
{
fl oat mat[row][col], s;
int i,j,r,c;
printf("\n Input number of rows:");
scanf("%d", &r);
printf("\n Input number of cols:");
scanf("%d", &c);
for(i = 0 ; i< r; i++)
{
for(j = 0 ;j<c; j++)
{
scanf("%f", &mat[i][j]);
}
}
printf("\n Entered 2D array is as follows:\n");
for(i = 0; i < r; i++)
{
for(j = 0; j < c; j++)
{
printf("%f", mat[i][j]);
}
printf("\n");
}
s = 0.0;
for(i = 0; i < r; i++)
{
for(j = 0; j < c; j++)
{
s += mat[i][j] * mat[i][j];
}
}
printf("\n Norm of above matrix is: %f", sqrt(s));
return 0;
}
```

C Program to read a matrix and find sum, product of all elements of two dimensional (matrix) array

```
include  <stdio.h>
#define MAXROW    10
#define MAXCOL    10
int main()
{
        int matrix[MAXROW][MAXCOL];
        int i,j,r,c;
        int sum,product;

        printf("Enter number of Rows :");
        scanf("%d",&r);
        printf("Enter number of Cols :");
        scanf("%d",&c);

        printf("\nEnter matrix elements :\n");
        for(i=0;i< r;i++)
        {
                for(j=0;j< c;j++)
                {
                        printf("Enter element [%d,%d] : ",i+1
                        scanf("%d",&matrix[i][j]);
                }
        }
        sum=0;
        product=1;

        for(i=0;i< r;i++)
        {
                for(j=0;j< c;j++)
                {
                        sum+=matrix[i][j];
                        product*=  matrix[i][j];
        }       }
printf("\nSUM of all elements : %d \nProduct of all elements :%d",sum,product);
        return 0;
}
```

```
Enter number of Rows :3
Enter number of Cols :3

Enter matrix elements :
Enter element [1,1] : 1
Enter element [1,2] : 1
Enter element [1,3] : 1
Enter element [2,1] : 2
Enter element [2,2] : 2
Enter element [2,3] : 2
Enter element [3,1] : 3
Enter element [3,2] : 3
Enter element [3,3] : 3

SUM of all elements : 18
Product of all elements :216
```

Find the sum of diagonal elements of a matrix

```
#include < stdio.h >
int main()
{
        int a[10][10],i,j,sum=0,r,c;
        clrscr();
        printf("\n Enter the number of rows and column ");
        scanf("%d%d",&r,&c);
        printf("\nEnter the %dX%d matrix",r,c);
        for(i=0;i < r;i++)
        {
                for(j=0;j < c;j++)
                {
                        scanf("%d",&a[i][j]);
                }//for
        }//for

        for(i=0;i < r;i++)
        { for(j=0;j < c;j++)
                {  if(i==j)
                        {
                                sum+=a[i][j];
                        }
                }//for
```

```
1  2  3
2  4  6
3  5  8

Sum of diagonal=13
```

```
        }//for
        printf("\nThe sum of diagonal elements is %d",sum); return 0;
}//main
```

Sum of rows and columns

```
#include <stdio.h>
 void main ()
 {
int array[10][10];
int i, j, m, n, sum = 0;
printf("Enter the order of the matrix\n");
scanf("%d %d", &m, &n);
printf("Enter the co-efficients of the matrix\n");
for (i = 0; i < m; ++i)
      {
         for (j = 0; j < n; ++j)
           {
              scanf("%d", &array[i][j]);
           }
        }
          for (i = 0; i < m; ++i)
        {
           for (j = 0; j < n; ++j)
           {
              sum = sum + array[i][j] ;
           }
            printf("Sum of the %d row is = %d\n", i, sum);
           sum = 0;
           }
        sum = 0;
        for (j = 0; j < n; ++j)
         {
            for (i = 0; i < m; ++i)
           {
              sum = sum + array[i][j];
           }
            printf("Sum of the %d column is = %d\n", j, sum);
           sum = 0;
        }
    }
```

Output

```
Enter the order of the matrix
2 2
Enter the co-efficients of the matrix
23 45

80   97
Sum of the 0 row is = 68
Sum of the 1 row is = 177
Sum of the 0 column is = 103
Sum of the 1 column is = 142
```

C Program to do the Sum of the Main & Opposite Diagonal Elements of a MxN Matrix

```
#include <stdio.h>
void main ()
    {           static int array[10][10];
      int i, j, m, n, a = 0, sum = 0;
        printf("Enetr the order of the matix \n");
      scanf("%d %d", &m, &n);
        if (m == n )
      {
         printf("Enter the co-efficients of the matrix\n");
        for (i = 0; i < m; ++i)
        {
           for (j = 0; j < n; ++j)
           {
              scanf("%d", &array[i][j]);
           }
        }
          printf("The given matrix is \n");
        for (i = 0; i < m; ++i)
        {
           for (j = 0; j < n; ++j)
           {
              printf(" %d", array[i][j]);
           }
           printf("\n");
        }

        for (i = 0; i < m; ++i)
        {
           sum = sum + array[i][i];
           a = a + array[i][m - i - 1];
        }
      printf("\nThe sum of the main diagonal elements is = %d\n", sum);
      printf("The sum of the off diagonal elements is  = %d\n", a);
        }
       else
        printf("The given order is not square matrix\n");
      }
```

```
Enter the order of the matix
2 2
Enter the co-efficients of the matrix
40 30
38 90
The given matrix is
 40 30
 38 90

The sum of the main diagonal elements is
= 130
The sum of the off diagonal elements is
= 68
```

C Program to Find the Frequency of Odd & Even Numbers in the given Matrix

```
#include <stdio.h>
  void main()
 {

          static int array[10][10];
          int i, j, m, n, even = 0, odd = 0;

          printf("Enter the order ofthe matrix \n");
          scanf("%d %d", &m, &n);

          printf("Enter the coefficients of matrix \n");
          for (i = 0; i < m; ++i)
      {
           for (j = 0; j < n; ++j)
           {
              scanf("%d", &array[i][j]);
              if ((array[i][j] % 2) == 0)
              {
                 ++even;
```

```
                }
                else
                   ++odd;
             }

           }

           printf("The
           given matrix is
           \n");for (i = 0; i
           < m; ++i)
       {
          for (j = 0; j < n; ++j)
          {
             printf(" %d", array[i][j]);
          }
          printf("\n");
       }
printf("\n The frequency of occurrence of odd number = %d
\n", odd);printf("The frequency of occurrence of even number =
%d\n", even);

   }
```



```
Enter the
order of the
matrix3 3
Enter the
coefficients of
matrix34 36 39
23 57 98
12 39 49
The
 giv
 en
 ma
 trix
 is
 34
 36
 39
 23 57 98
 12 39 49
```

The frequency of occurrence of odd number = 5The frequency of occurrence