Array:

→ An array can be defined as an ordered list of homogeneous data elements.

→ These elements may be of type int, float, char or double.

→ All the elements in array are stored in consecutive memory location

→ An array is described by a single name and each individual data item in the array is referenced by a subscript (or index) enclosed in a pair of square bracket.

→ This subscript indicates the position of an individual data item in an array

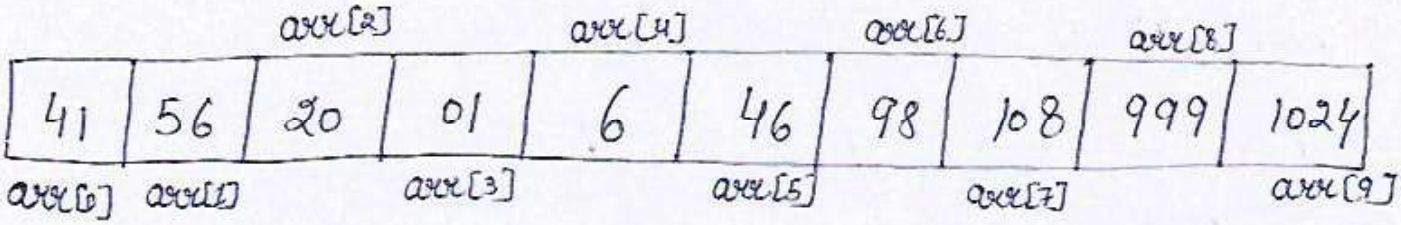
→ In C language, the subscript value ranges from 0 to one less than the maximum size. If the size of an array is 10, Then, the first subscript is 0, the second subscript is 1 & so on. The last subscript is 9. In general, the i^{th} element has subscript $(i-1)$.

eg int arr[10];

arr[0], arr[1], arr[2] - - - - - arr[9].

Memory Representation of Array

int arr[10];



Declaration of Array

It is a linear list of fixed number of data items of same type. All these data items are accessed using the same name using a single subscript.

Syntax

datatype arrayname [size];

Where

datatype → type of data stored in array

arrayname → name of array

size → maximum no. of elements that can be stored in an array.

Total Memory size of Array

Total size = size * (size of data type)

where:

size → number of elements in one dimensional array

size of datatype → memory requirement of data type.

Qe Program to accept n integers & store them in an array called num. Also print them

```
#include<stdio.h>
void main()
    {
      int n, i, num[20];
      printf("Enter the size of an array\n");
          scanf("%d", &n);
      for(i=0; i<n; i++)
       {
          scanf("%d", &num[i]);
       }
```

```
printf ("Array is \n");
    for (i=0; i<n; i++)
      {
        printf ("num [%d] = %d \n", i, num[i]);
      }
  }
```

Program to Read marks of 5 Subjects in array:

```
# include <stdio.h>
void main()
    {
      int i;
      float marks [5];
    for (i=0; i<=4; i++)
      {
        scanf ("%f", & marks[i]);
      }
```

Program (Array)

Qo1 Program to arrange elements of array in increasing order.

```
# include <stdio.h>
  Void main ()
  {
     float array [20];
      int i, j, n, temp;
    printf (" Enter the size of an array\n");
     scanf ("%d", &n);
     printf ("Enter elements \n");
       for (i=0; i<n; i++)
           scanf ("%f", & array[i]);
      printf (" Input array is \n");
          for (i=0; i<n ; i++)
              printf ("%f\n", array [i]);
       for (i=0; i<n-1; i++)
          {
             for (j= i+1 ; j<n; j++)
          {
             if (array [i] >= array [j])
```

```
{
    temp = array[i];
    array[i] = array[j];
    array[j] = temp;
  }
 }
}

printf("Sorted array is...\n");
    for(i=0; i<n; i++)
        printf("%d\n", array[i]);
  }
```

Initializing a One dimensional Array

→ Initializing means assigning some values to the variable that undergoes processing.

→ Like other variables, elements of array can also be initialized. All initialized values should be constants.

→ It is possible to declare all the variables of array during its declaration

Syntax

datatype array_name [size] = {element1, element2 ... elementn};

where

datatype → data type of elements stored

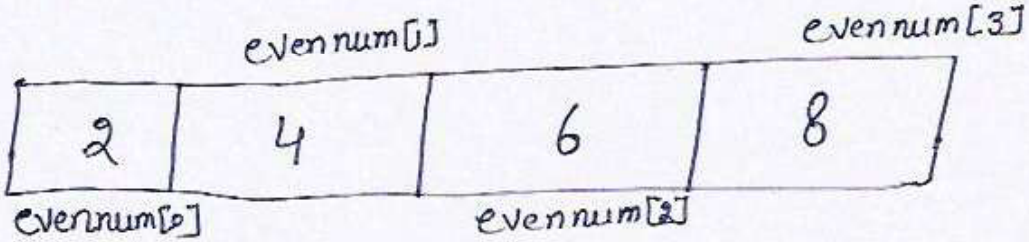
array_name → name of array

size → maximum number of elements

element1, element2 ... → initial value of data elements

eg

```
int evennum[4] = {2, 4, 6, 8};
```



Note: If no value is explicitly assigned to any subscript of array it takes zero as initialised value.

→ It is also possible to declare arrays with their initial values, without writing any value within brackets.

eg int age[ ] = {16, 26, 18, 24, 23, 52}

here

age[0] = 16
age[1] = 26
age[2] = 18
age[3] = 24
age[4] = 23
age[5] = 52

Program

```
# include <stdio.h>
main()
  {
    int marks[ ] = {45, 65, 90, 87, 74};
   int i;
   printf("Elements of array are\n");
   for(i=0; i<5; i++)
     printf("marks[%d] = %d \n", i, marks[i]);
  }
```

6b Program to accept two integer array & find sum of corrosponding elements of these array.

```
#include <stdio.h>
void main()
{
 int arr1[10], arr2[10], sum[10];
 int i, n;
 printf("Enter size of arrays \n");
 scanf("%d", &n);
 printf("Enter elements of array 1 \n");
 for(i=0; i<n ; i++)
    scanf("%d" & arr1[i]);
 printf("Enter elements of array2 \n");
    for(i=0; i<n ; i++)
      scanf("%d", & arr2[i]);
 for(i=0; i<n; i++)
   sum[i] = arr1[i] + arr2[i];
 printf("Sum of elements of array1 & array2 \n");
   for(i=0; i<n; i++)
   printf("%d\n", sum[i]);
```

Q. Program to print largest element of an array size 10;

```
#include <stdio.h>
void main()
{
    int x[10], large, i;
    printf("Enter elements of array\n");
    for (i=0; i<10; i++)
        scanf("%d", &x[i]);
    large = x[0];
    for (i=1; i<10; i++)
    {
        if(x[i] > large)
            large = x[i];
    }
    printf("largest element is %d", large);
}
```

Q. Program to search desired element in n sized array

```
#include <stdio.h>
void main()
{
  int n, i, found=0, listn[20], key_element;
 printf("Enter number of elements in array\n");
  scanf("%d", &n);
  printf("Enter %d numbers\n", n);
   for(i=0; i<n; i++)
     scanf("%d", &listn[i]);
  printf("Enter key element to search\n");
    scanf("%d", &key_element);
   for(i=0; i<n; i++)
    {
      if(listn[i]==key_element)
        found=1;
    }
   if(found==1)
     printf("Element is found");
  else
     printf("Element is not found");
}
```

Program to compute sum of even & odd numbers stored in an array.

```
#include <stdio.h>
void main()
{
    int num[20];
    int i, esum=0, odsum=0, n;
    printf("Enter size of array \n")
    scanf("%d", &n);
    printf("Enter the elements of array \n");
    for(i=0; i<n; i++)
        scanf("%d", &num[i]);
    for(i=0; i<n; i++)
    {
        if(num[i]%2==0)
            esum += num[i];
        else
            odsum += num[i];
    }
    printf("Sum of even numbers is %d", esum);
    printf("Sum of odd numbers is %d", odsum);
}
```

Multidimensional Array :

If the number of subscript is more than one then such array is called multidimensional array.

eg - 2D (two dimensional)
3D (three dimensional)

Two Dimensional Array

2D array is an ordered table of homogeneous elements. It is generally, reffred to as matrix of some rows & some columns.

Syntax

```
datatype array-name [rows] [columns];
```

where

datatype → types of data stored in array
array_name → name of array
row → number of elements to be processed under subscript 1
Column → number of elements to be processed under subscript 2

eg. int marks [5] [3];

In this example array has 5 rows & 3 columns.

Structure of array is like:

marks [0] [0] =	marks [0] [1] =	marks [0] [2] =
marks [1] [0] =	marks [1] [1] =	marks [1] [2] =
marks [2] [0] =	marks [2] [1] =	marks [2] [2] =
marks [3] [0] =	marks [3] [1] =	marks [3] [2] =
marks [4] [0] =	marks [4] [1] =	marks [4] [2] =

Initialization of 2-D Array

The elements of two dimensional can also be initialized at the time of declaration

Syntax

datatype arrayname [size1] [size2] = {e1, e2 ... en};

where → datatype → Type of data elements in array
arrayname → name of array
e1, e2 .. en → initial values to be assigned to n elements of array

eg int matrix [3] [3] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

matrix [0] [0] = 1 matrix [0] [1] = 2 matrix [0] [2] = 3

matrix [1] [0] = 4 matrix [1] [1] = 5 matrix [1] [2] = 6

matrix [2] [0] = 7 matrix [2] [1] = 8 matrix [2] [2] = 9

Qo Program to Read & Print elements of a Matrix.

```
# include <stdio.h>
  Void main ( )
      {
        int n, i, j, row, col;
        int mat [10] [10];
       printf (" Enter order of matrix \n");
        scanf ("%d %d", &row, & col);
        printf (" Enter elements of array (matrix) \n");
        for (i=0; i< row ; i++)
          { for (j=0; j<col; j++)
            { scanf ("%d", & mat [i] [j]);
            }
          }
```

```
printf("Matrix is \n");
  for(i=0; i<row; i++)
     {
       for(j=0; j<col; j++)
         {
           printf("%d", mat[i][j]);
            printf("\t");
         }
       printf("\n");
     }
}
```

Output

```
Enter order of matrix
  3  3
Enter elements of array (matrix)
1
2
3
4
5
6
7
8
9
```

Matrix is

```
1  2  3
4  5  6
7  8  9
```

~~Matrix is~~
~~1 2 3~~

Q. Program to accept two matrix of same order & find the sum of these matrix & print sum

```
# include <stdio.h>
void main()
    {
      int n, i, j, row, col;
      int mat1 [10] [10], mat2 [10] [10], matsum [10] [10];
      printf (" Enter Rows & Columns \n");
       scanf ("%d %d", & row, & col);
      printf (" Enter elements of matrix 1 \n");
      for (i=0; i< row ; i++)
          {
            for (j=0; j< col; j++)
              {
                scanf ("%d", & mat1[i] [j]);
              }
          }
      printf (" Enter elements of matrix 2 \n");
          for (i=0; i< row; i++)
              {
                for (j=0; j< col; j++)
                {
                  scanf ("%d", & mat2 [i] [j]);
                }
              }
```

```
for (i=0; i< row ; i++)
  {
    for (j=0; j<col; j++)
      {
        matsum[i][j] = mat1[i][j] + mat2[i][j];
      }
  }
printf (" The sum matrix is..
for (i=0; i< row; i++)
  {
    for (j=0; j< col ; j++)
      { printf ("%d", matsum[i][j]);
      }
      printf ("\n");
  }
}
```

Program to find product of two matrices & print the product matrix.

```
#include <stdio.h>
void main()
{
  int i, j, k, row1, col1, row2, col2;
  int matA[10][10], matB[10][10], prodmat[10][10];
  printf("Enter order of matrix 1\n");
  scanf("%d %d", &row1, &col1);
  printf("%d %d", &
  printf("Enter order of matrix 2\n");
  scanf("%d %d", &row2, &col2);
  if(col1 == row2)
  {
    printf("Enter the elements of Matrix 1\n");
    for(i=0; i<row1; i++)
    {
      for(j=0; j<col1; j++)
      {
        scanf("%d", &matA[i][j]);
      }
    }
  printf("Enter elements of Matrix 2\n");
```

```
for (i=0; i<row2; i++)
  {
     for (j=0; j<col2; j++)
       { scanf ("%d", &matB[i][j]);
       }
  }

for (i=0; i<row1; i++)
  {
     for (j=0; j<col2; j++)
       {
          prodmat[i][j]=0 ;
          for (k=0; k<row2; k++)
            {
          prodmat[i][j]= prodmat[i][j]+ (matA[i][k] *
                                          matB[k][j]);
          }
       }
  }
printf (" Product matrix is ... \n");
    for(i=0; i<row1; i++)
       { for (j=0; j<col2; j++)
          {
            printf ("%d", prodmat[i][j]);
          }
            printf ("\n");
       }
  }
else
  printf (" Multiplication is Not Possible \n");
}
```