Date / /

UNIT = 1

Chapter = 1

[ARRAYS]

The variables that we have till now are capable of storing only one value at a time.
Consider a situation when we want to store and display the age of 100 employes. for this, we have to do the following :-

(i) Declare 100 different variables to store the age of the employees.
(ii) Assign a value to each variable.
(iii) Display the value of each variable.

It would be very difficult to handle so many variables in the program and the program would become very lengthy. The concept of array is useful in these type of situation, where we can group similar type of data items.

Array :-
An array is a collection of similar type of data itmes and each data item is called an element of the array. The datatype of elements may be any valid data type like :- int, Float, char.

The elements of array share the same variable but each element has a different index number known as subscripts.
For about the problems we can take array variable age [100] of type int.

Date / /

The size of this array variable is 100. so it is capable of storing 100 integer value. The individual elements of this array are:-

age[0], age[1], age[2], age[3], ---------- age[99]

In C, the subscripts starts from 0.
so, age[0] is first element and age[1] is second element.

Array can be single dimensional or multi-dimensional. The number of subscripts determine the dimension of the array.

1-D array has one subscripts.
2-D array has two subscripts.
and so on.

- The one dimensional arrays are known as vector.
- The two dimensional arrays are known as Matrices.

V. Imp ★ 1st Sess

One Dimensional Array :→

1. Declaration of 1-D Array :-

Syntax for declaration of an array is :-

datatype Array_name[array size];

Here, Array_name denotes the name of the array and it can be any valid C Identifier.
data-type is the datatype of elements of array.
Here some examples of array declaration :-

```
int age[100];
float sal[15];
char grade[20];
```

datatype, array_name[size];

int → datatype
marks → array_name (variable)
[50] → size

Date / /

When the array is declare the compiler allocate space in memory to hold whole the elements of the array. so, the compiler should know the size of array and at the compile time.

Accessing 1-D Array Element :-

The elements of an array can be access by the specifying the array name followed by sub-scripts in brackets.

In C the array subscript start from 0, hence if based an array of size 5 then the valid subscript will be shown 0 to 4. The last valid subscript 1 less then the size of array.

This last valid subscript in sometime known as the upper bound of the array and '0' is known as the lower bound of the array.

Let us take an array :-

```
int arr[5];
```

Size of array is 5 can hold 5 integer elements
The element of this array :-

arr[0], arr[1], arr[2], arr[3], arr[4]

↓ Lower bound — ↓ Upper bound

Processing 1-D Arrays :-

For processing array, we generally use a for loop and the loop variable is huge at the place of sub-script. The initial value of loop variable is taken '0'.

Since, array subscript start from '0'. The loop variable is increase by 1 each time. so, that we can access and process the next element in the array.

Note :- The total no. of passes in the loop will be equal to the no. of elements in the array. is in each pass, we will process one element.

Example :- Suppose, arr[10] is an array of int type.

① Reading value in arr[10] :-

```
for(i=0; i<10; i++)
scanf("%d", &arr[i]);
```

② Displaying the value of arr[10] :-

```
for(i=0; i<10; i++)
printf("value of array %d", arr[i]);
```

1st Sess

Program 1 :- Write a program to input values into an Array and display them :-

```
#include<stdio.h>
#include<conio.h>
  void main()
  {
    clrscr();
  int arr[5], i;
  for(i=0; i<5; i++)
  {
  printf("Value of array :%d", arr[i]);
```

Date / /

```
printf("Enter the value of array:");
scanf("%d", &arr[i]);
}
for(i=0; i<5; i++)
  {
    printf("Value of array:%d", arr[i]);
    printf("\n");
    }
      getch();
      }
```

Output:-

Enter the value of array: 1
Enter the value of array: 2
Enter the value of array: 3
Enter the value of array: 4
Enter the value of array: 5

Value of array : 1
Value of array : 2
Value of array : 3
Value of array : 4
Value of array : 5

1st scss.

Program 2:- Write a program to add the element of an Array :-

```
#include<stdio.h>
#include<conio.h>
  void main()
  {
    clrscr();
  int i, sum=0, arr[10];
```

Date / /

```
for(i=0; i<5; i++)
{
printf("Enter the value of array:");
scanf("%d", &arr[i]);
}
sum+=arr[i];
printf("The sum of an Array:%d \n", arr[i] sum);
printf("\n");
getch();
}
```

Output :- Enter the value of array: 1
Enter the value of array: 2
Enter the value of array: 3
Enter the value of array: 4
Enter the value of array: 5

Value of Array: 15

* Adding All the Elements of arr[10] :-

```
sum = 0;
for(i=0; i<10; i++)
sum+=arr[i];
```

1st Sess ✓ ★ Two 2 Dimensional Array :→

1. Declaration and associating individual element of 2D array :-

The Syntax of declaration of 2D array is similar to that of 1D Array but here we have 2 subscripts :-

```
datatype array_name [row size] [column size];
```

Date / /

datatype array-name [row size] [column size];

Here, row size specify the number of rows and column size specify the number of column in array.

☆ The total no. of element in the array are
row size * column size

For Example :- int arr[4] [5];

Here arr[4] [5] is a 2D array with 4 rows and 5 columns.
first element of this array → arr[0] [0]
last element of this array → arr[3] [4]

	col 0	col 1	col 2	col 3	col 4
row 0	arr [0] [0]	arr [0] [1]	arr [0] [2]	arr [0] [3]	arr [0] [4]
row 1	arr [1] [0]	arr [1] [1]	arr [1] [2]	arr [1] [3]	arr [1] [4]
row 2	arr [2] [0]	arr [2] [1]	arr [2] [2]	arr [2] [3]	arr [2] [4]
row 3	arr [3] [0]	arr [3] [1]	arr [3] [2]	arr [3] [3]	arr [3] [4]

Processing 2-D Array Elements :-
For processing 2-D array we use two for loops the outer for loop correspond to row and inner for loop correspond to columns.
int arr[4] [5];

i) Taking Input in arr :-

```
for(i=0; i<4; i++)
for(j=0; j<5; j++)
```

```
{
scanf("%d", &arr[i][j]);
}
```

3. Display value of arr :-

```
for(i=0; i<4; i++)
 for(j=0; j<5; j++)
 {
  printf("%d", arr[i][j]);
 }
```

Program based on 1-D Array :-

Program 3:- A Program to input an array and count even and odd numbers :-

```
#include<stdio.h>
#include<conio.h>
 void main()
{
 int arr[10], i, even=0, odd=0;
printf("Enter ten numbers to check:");
 for(i=0; i<10; i++)
  {
  scanf("%d", &arr[i]);
  if(arr[i]%2==0)
  {
  even++;
  }
  else
   {
   odd++;
   }
   }
```

Date / /

```
printf("Number of Even no. are : %d\n Number of odd no. are : %d\n
        getch();
    }
```

Output :- Enter ten numbers to check: 1
2
3
4
5
6
7
8
9
10
Number of Even no. are: 5
Number of Odd no. are: 5

Program based on 2-D Array :-

1st sess.

Program 4 :- A Program to input and display a matrix :-

```
#include<stdio.h>
#include<conio.h>
void main()
{
  clrscr();
 int mat[3][3], i, j;
printf("Enter nine values of matrix:");
    for(i=0; i<3; i++)
    for(j=0; j<3; j++)
   {
     scanf("%d", &mat[i][j]);
   }
```

Date / /

```
printf("Matrix entered by user is:");
    for(i=0; i<3; i++)
     for(j=0; j<3; j++)
{
    if(i==1 && j==0)
    printf("\n\n");
    if(i==2 && j==0)
    printf("\n\n");
    printf("%d", mat[i][j]);
    }
      getch();
       }
```

Output :- Enter elements of 3*3 matrix:

or

Enter nine values of matrix: 1

2

3

4

5

6

7

8

9

Matrix entered by user is: 1 2 3

4 5 6

7 8 9

Date / /

1st sess.

Program 5:- A program for addition of two matrices:-

```
#include<stdio.h>
#include<conio.h>
  void main()
{
  clrscr();
int  i,j, mat1[3][3], mat2[3][3], mat3[3][3];
printf("Enter nine elements of first matrix:");
  for(i=0; i<3; i++)
  for(j=0; j<3 ; j++)
   {
    scanf("%d", &mat1[i][j]);
   }
printf("Enter the nine elements of second matrix:");
  for(i=0; i<3; i++)
  for(j=0; j<3 ; j++)
{
 scanf("%d", &mat2[i][j]);
}
  printf("sum:");
   for(i=0; i<3; i++)
   for(j=0; j<3; j++)
  {
    mat3[i][j] = mat1[i][j] + mat2[i][j];
   if(i==1 && j==0)
   printf("\n\n");
   if(i==2 && j==0)
   printf("\n\n");
printf("\t%d", mat3[i][j]);
   }
      getch();
   }
```

Date / /

Output :- Enter Nine elements of fist matrix : 1
2
3
4
5
6
7
8
9
Enter the nine elements of second Matrix: 2
4
6
8
10
12
14
16
18

Sum: 3 6 9
12 15 18
21 24 27

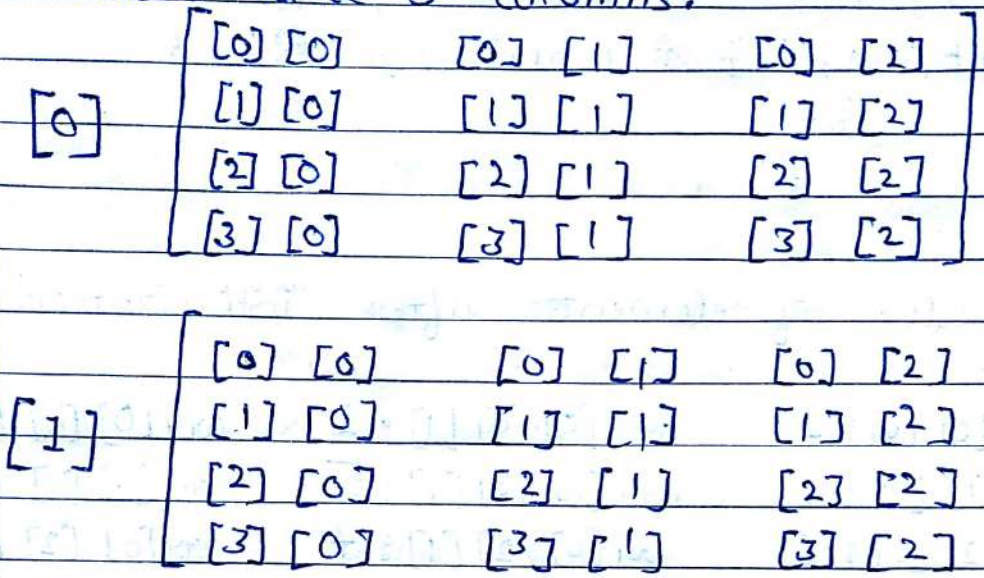
1st sess.

☆ Multi-Dimensional Array :→
Arrays with more than two Dimensions :-
We will just give a brief or over view 3-D arrays. We can think of 3D array as an array of 2-D array. for example, if we have an array,

int arr[2] [4] [3];

Date / /

We can think of this array which consist of 2-D array and each of those 2D array have 4 rows and 3 columns.



The individual element are :→

arr[0][0][0], arr[0][0][1], arr[0][0][2], arr[0][1][0] ..... arr[0][3][2]

arr[1][0][0], arr[1][0][1], arr[1][0][2], arr[1][1][0] ..... arr[1][3][2]

Total no. of elements in above array :-

$2 * 4 * 3 = 24$

This array can be initialized as :-

```
int arr[2][4][3]
  {
{1,2,3} * matrix 0, Row 0
{4,5} * matrix 0, Row 1
{6,7,8} * matrix 0, Row 2
{9}   * matrix 0, Row 3
   }

{
```

Date / /

{10, 11} * matrix 1, Row 0
{12, 13, 14} * matrix 1, Row 1
{15, 16} * matrix 1, Row 2
{17, 18, 19} * matrix 1, Row 3
}
}

The value of elements after initialization are as :-

[0]
arr[0][0][0] : 1 arr[0][0][1] : 2 arr[0][0][2] : 3
arr[0][1][0] : 4 arr[0][1][1] : 5 arr[0][1][2] : 0
arr[0][2][0] : 6 arr[0][2][1] : 7 arr[0][2][2] : 8
arr[0][3][0] : 9 arr[0][3][1] : 0 arr[0][3][2] : 0

[1]
arr[1][0][0] : 10 arr[1][0][1] : 11 arr[1][0][2] : 0
arr[1][1][0] : 12 arr[1][1][1] : 13 arr[1][1][2] : 14
arr[1][2][0] : 15 arr[1][2][1] : 16 arr[1][2][2] : 0
arr[1][3][0] : 17 arr[1][3][1] : 18 arr[1][3][2] : 19

1st sess.

1-D Array and Functions :→

1. Passing indivisual array and function :-

We know that array element is treated created as any other simple variable any other program so we can pass indivisual array elements as arguement to a function like other simple variables.

3rd sess

Program 6 :- Write a program to pass array elements to a function :-

```
#include<stdio.h>
#include<conio.h>
```

Date / /

```
int check(int num);
int main()
{
clrscr();
int arr[10], i;
for(i=0; i<10; i++)
{
printf("Enter a number to check:");
scanf("%d", &arr[i]);
check(arr[i]);
}
getch();
}
int check(int num)
{
if(num % 2 == 0)
printf("No. is even\n\n");
else
printf("No. is odd\n\n");
}
```