:Unit-2 = Strings :-

M T W T F S S
Page No:
Date:
YOUVA

=) string is a sequence of characters terminated with a null character '\0'. strings are defined as an essay of characters

* Declaration of strings :-

=) Declaring a string is as simple as declaring a one-dimensional array. Below is the basic syntax for declaring a string

Syntax:-

```
char string_name[size];
```

string_name: ↑ Variable Name

size: ↑ length of string

* Initializing a string:-

- A string can be initialized in different ways.
4 ways to initialize a string in C.

1. Assigning a string literal without size:-

```
char string[] = "Hello_world";
```

- string literals can be assigned without size

2. Assigning a string literal with a predefined size.

- string literals can be assigned with a pre-defined size:-

```
char string[50] = "Hello_world";
```

3. Assigning character by characters with size:-

```
char string[14] = {'h','e','l','l','o','w','o','r','l','d'};
```

4. Assigning character by characters without size:-

```
char string[] = {'h','e','l','l','o','w','o','r','l','d'};
```

:- Chap-1 - Array :-

M T W T F S S
Page No.:
Date: 22-02-23
YOUVA

* Topics :-

(i) Introduction of array.
(ii) Initialization of array.
(iii) Types of array.
(iv) - 1-D array (one-dimensional)
- 2-D array
- Multi-dimensional or 3-D array.

(iv) Array - operations.
- Insertion.
- Deletion.
- Searching.
- Merging.
- Sorting

(v) Operations on 2-D array
- Matrices addition. / Multiplication (3×3) or (2×2)

* Array :-

⇒ An array is collection of different elements which have same name & datatype with fix size.

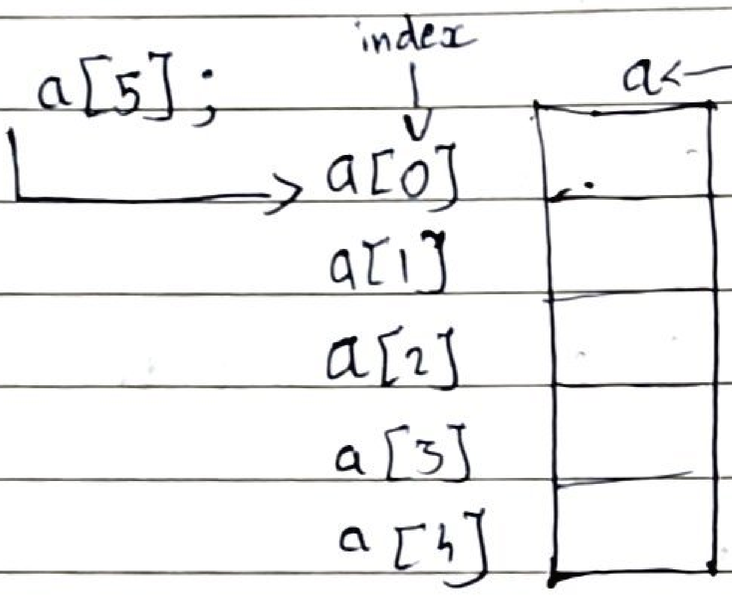
⇒ An array is contiguos memory location.
⇒ It is a linear data structure.
⇒ The index (position) of an array ~~is~~ always starts with zero (0).
⇒ Considering that if we have array n (arr), the last index of that array should be 'n-1'.

* Syntax :-

datatype array_name [size] ;

E.g.,

(i) int a[5];



(ii) int QB[63];

E.g.

* to take elements of array from user:-

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

void main ()
    {
        int a[5], i;
        printf("Enter the elements \n");
        for (i=0; i<5; i++)
            {
            scanf("%d", &a[i]);
            }
        getch();
    }
```

Output:

Output:

Enter the elements

30

15

10

20

25

Initialization of Array :-

⇒ The Initialization of array is done by using index of each elements.

Method - 1 :-

```
int a[0] = 10
int a[1] = 15
int a[2] = 20
int a[3] = 25
int a[4] = 30
```

Method - 2 :

```
int a[5] = {10,15,20,25,30};
```

Eg. Initialization of array at compile time :

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

void main ()
{
    int a[5] = {10,15,20,25,30};
    printf ("Elements are: \n");
    for (i=0;i<5;i++)
    {
        printf ("a[%d] = %d", i, a[i]);
    }
```

```
    getch();
}
```

Output:

Elements are:

10
15
20
25
30

1-03-23

* Types of Array:

- The array which have only one single index is known as 1-dimensional array.

Syntax:

```
datatype array_name[size];
```

Eg:- int a[5];

H.W Program:-

output:- Enter elements

11
21
31
42
52

Entered elements are

11
21
31
42
52

~~2-dimensi~~

2-dimensional array:

=) This array is used to store a data in a tabular form.

- Tabular form contains row and columns. A 2-D array generally creates matrices form. ① 2x2 $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

② 3x2 $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$

② 3x3 $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

③ 2x3 $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$

Syntax:-

datatype array_name[row size] [column size];

E.g.:-

```
int a[2][2]; // 2x2        int [2][3]; // 2x3
int a[3][3]; // 3x3
```

program:-

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

void main()
    {
      int a[2][2],i,j;
      printf("Enter elements:\n");
      for(i=0;i<2;i++)
       {
        for(j=0;j<2;j++)
        {
         scanf("%d",&a[i][j]);
        }
       }
     for(i=0;i<2;i++)
       {
        for(j=0;j<2;j++)
        {
         printf("a[%d][%d]=%d",i,j,a[i][j]);
        }
         printf("\n");
       }
         getch();
    }
```

Output:-

```
Enter elements:
1
5
10
15
```

a[0][0]=1 a[0][1]=5

a[1][0]=5 a[1][1]=15

M T W T F S S

Page No.:

Date:

YOUVA

3 or Multi-dimensional array :-

=> The array which have more than one or two dimension is called multi or 3 dimensional array.

=> This array is also called n-D. array

Syntax:-

datatype array_name[d1][d2][d3][d4]-----[dm];

Example:- int a[2][2][2];

Program:-

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

void main()
  {
    int a[2][2][2],i,j,k;
    printf("Enter elements:\n");
    for(i=0;i<2;i++)
      {
        for(j=0;j<2;j++)
        {
          for(k=0;k<2;k++)
          {
            scanf("%d",&a[i][j][k]);
          }
        }
      }
    for(i=0;i<2;i++)
      {
```

```
for (j=0; j<2; j++)
  {
    for (k=0; k<2; k++)
      {
        printf ("a[%d][%d][%d]=%d", i, j, k, a[i][j][k]);
      }
  }
}
getch();
}
```

* Output :-

* Operations on 1-D array :-

(i) Insertion
(ii) Deletion
(iii) Searching
(iv) Merging
(v) Sorting

(i) Insertion :-

This operation is used to insert an element in the array.

(ii) Deletion :-

This operation is used to delete an element from the array.