Ch: 4

ARRAY

An array is defined as the collection of similar type of data items stored at contiguous memory locations.(imp -1mark)

Arrays are the derived data type in C programming language which can store the primitive type of data such as int, char, double, float, etc. It also has the capability to store the collection of derived data types, such as pointers, structure, etc.

The array is the simplest data structure where each data element can be randomly accessed by using its index number.

C array is beneficial if you have to store similar elements. For example, if we want to store the marks of a student in 6 subjects, then we don't need to define different variables for the marks in the different subject.

Instead of that, we can define an array which can store the marks in each subject at the contiguous memory locations.

Type of C Array

- Array can be of following types:
  (i) One dimension array (1 - D)
  (ii) Two dimension array (2 - D)
  (iii) Multi dimension array (M – D)

How to declare array:- [1 mark]

- Array can be declare at design time and at run time.
- To declare array , C language provide 3 criteria's:
  (i) Name of Array
  (ii) Size of Array
  (iii) Type of Array

Ch: 4

Advantage of C Array

1) Code Optimization: Less code to the access the data.

2) Ease of traversing: By using the for loop, we can retrieve the elements of an array easily.

3) Ease of sorting: To sort the elements of the array, we need a few lines of code only.

4) Random Access: We can access any element randomly using the array.

Disadvantage of C Array

1) Fixed Size: Whatever size, we define at the time of declaration of the array, we can't exceed the limit. So, it doesn't grow the size dynamically like LinkedList which we will learn later.

How to declare Array in C

```
int num[35];  /* An integer array of 35 elements */
char ch[10];  /* An array of characters for 10 elements */
```

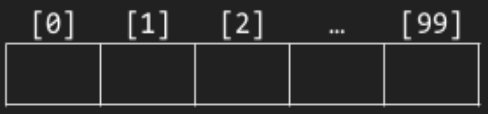
Similarly an array can be of any data type such as double, float, short etc.

Ch: 4

Definition: Array

- An array is a fixed size sequential collection of elements of same data type grouped under single variable name.

```
int rollno[100];
```



Fixed Size	Sequential	Same Data type	Single Name
Here, the size of an array is 100 (fixed) to store rollno	It is indexed to 0 to 99 in sequence	All the elements (0-99) will be integer variables	All the elements (0-99) will be referred as a common name rollno

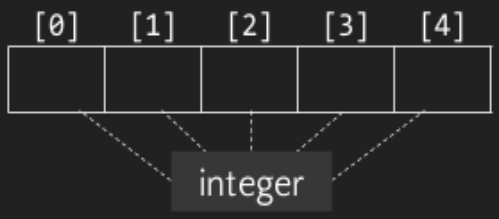
Declaring an array

Syntax

```
data-type variable-name[size];
```

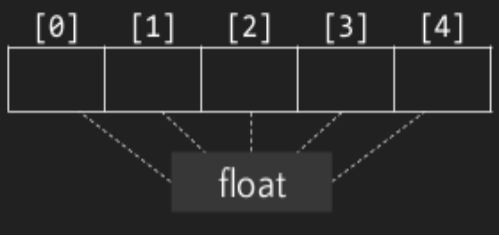
Integer Array

```
int mark[5];
```



Float Array

```
float avg[5];
```



- By default array index starts with `0`.
- If we declare an array of size 5 then its index ranges from `0` to 4.
- First element will be store at `mark[0]` and last element will be stored at `mark[4]` not `mark[5]`.
- Like integer and float array we can declare array of type `char`.

Ch: 4

Initialing and Accessing an Array

Declaring, initializing and accessing single integer variable

```
int mark=90;      //variable mark is initialized with value 90
printf("%d",mark); //mark value printed
```

Declaring, initializing and accessing integer array variable

```
int mark[5]={85,75,76,55,45}; //mark is initialized with 5 values
printf("%d",mark[0]); //prints 85
printf("%d",mark[1]); //prints 75
printf("%d",mark[2]); //prints 65
printf("%d",mark[3]); //prints 55
printf("%d",mark[4]); //prints 45
```

	[0]	[1]	[2]	[3]	[4]
mark[5]	85	75	65	55	45

Develop a program to count number of positive or negative number from an array of 10 numbers.

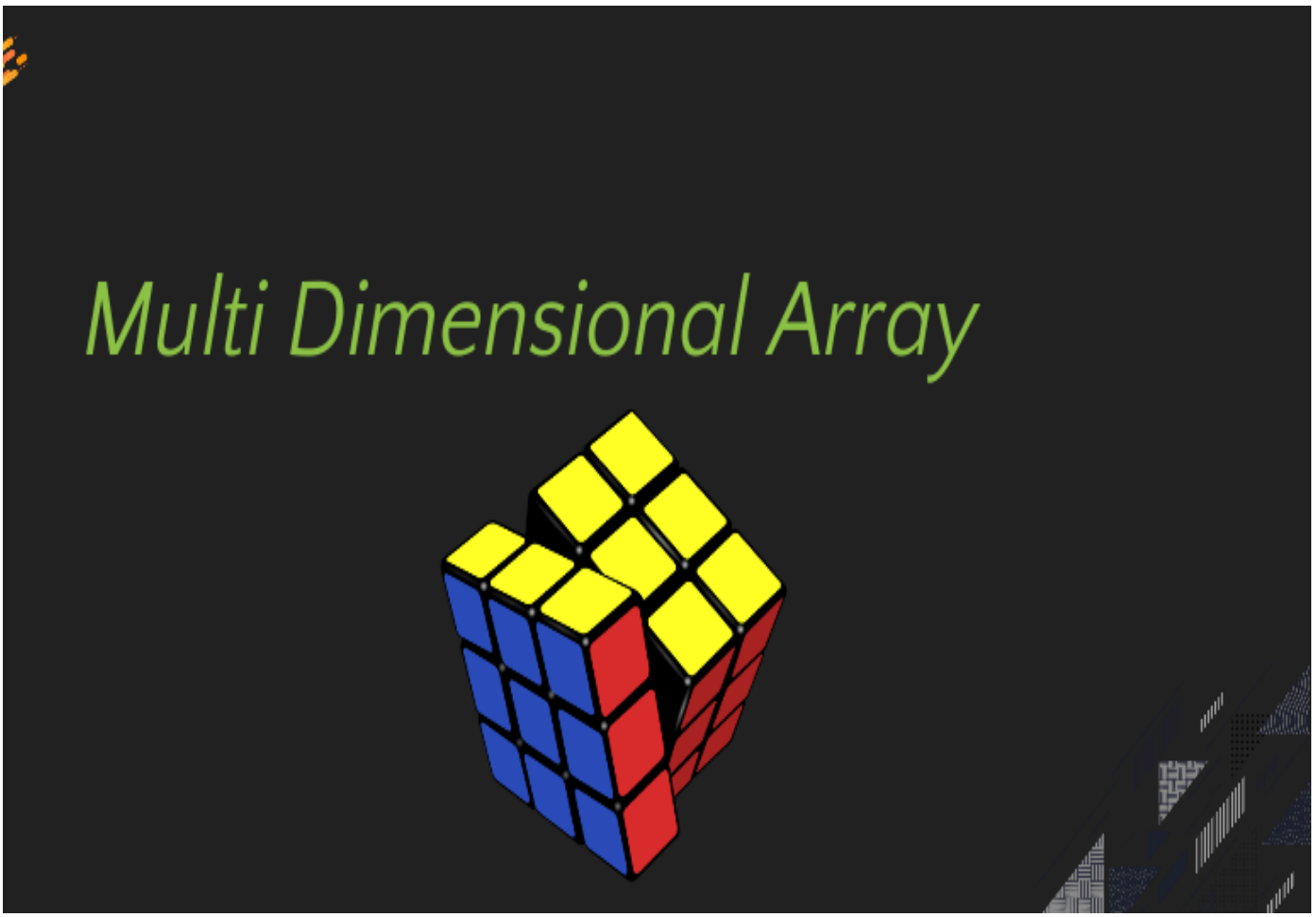
Program

```
1  void main(){
2      int num[10],i,pos,neg;
3      pos = 0;
4      neg = 0;
5      for(i=0;i<10;i++)
6      {
7          printf("Enter array element=");
8          scanf("%d",&num[i]);
9      }
10     for(i=0;i<10;i++)
11     {
12         if(num[i]>0)
13             pos=pos+1;
14         else
15             neg=neg+1;
16     }
17     printf("Positive=%d,Negative=%d",pos,neg);
18 }
```

Output

```
Enter array element=1
Enter array element=2
Enter array element=3
Enter array element=4
Enter array element=5
Enter array element=-1
Enter array element=-2
Enter array element=3
Enter array element=4
Enter array element=5
Positive=8,Negative=2
```

Ch: 4



Radha Ranpara

Ch: 4

Two dimensional/multidimensional Arrays

C language supports multidimensional arrays also. The simplest form of a multidimensional array is the two- dimensional array. Both the row's and column's index begins from 0.

Two-dimensional arrays are declared as follows,

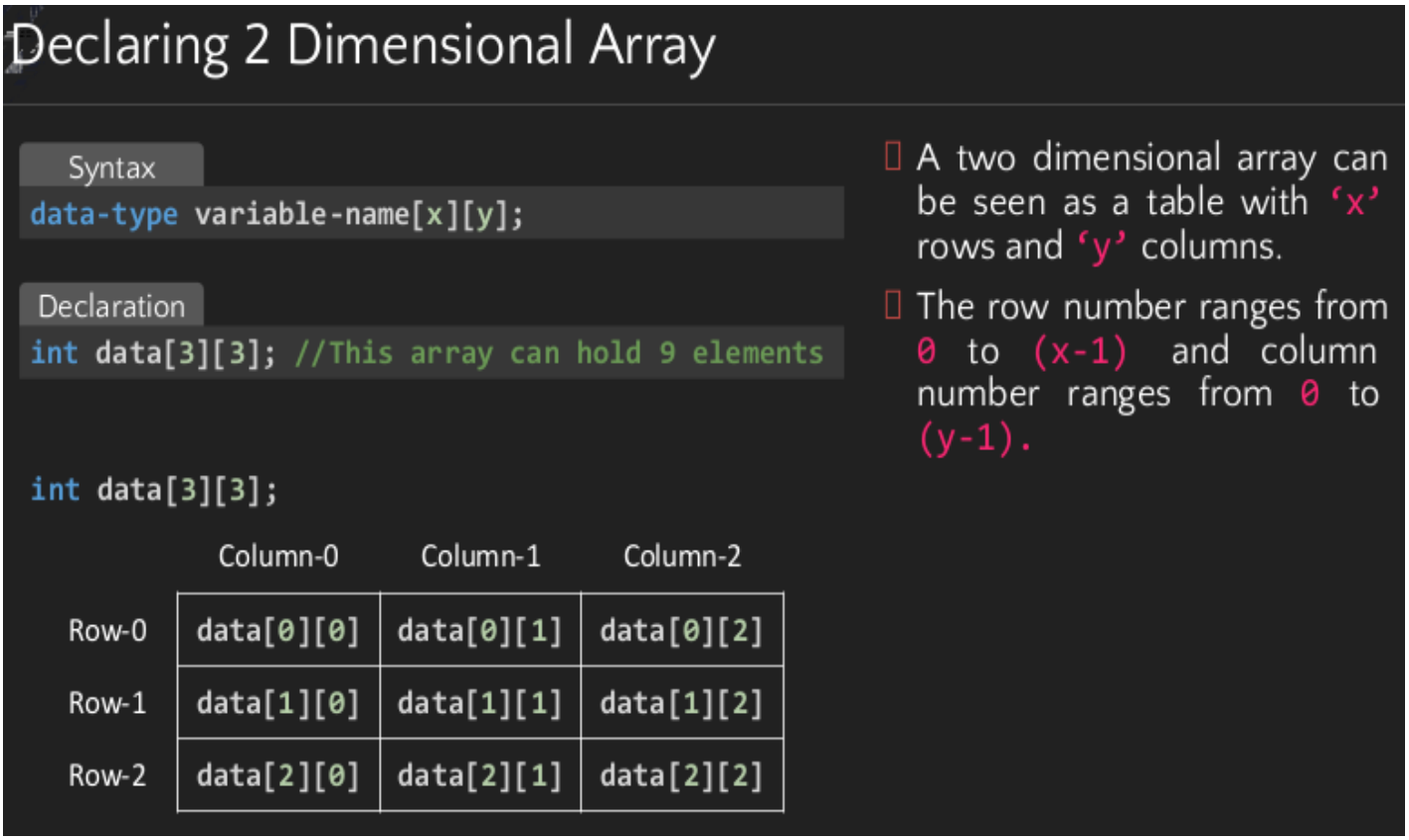
data-type array-name[row-size][column-size]

/* Example */

int a[3][4];

An array can also be declared and initialized together. Forexample,

int arr[][3] = {{0,0,0},{1,1,1}};



Ch: 4

Initialing and Accessing a 2D Array: Example-1

Program

```c
int data[3][3] = {
{1,2,3}, //row 0 with 3 elements
{4,5,6}, //row 1 with 3 elements
{7,8,9}  //row 2 with 3 elements
    };
printf("%d",data[0][0]); //1
printf("%d",data[0][1]); //2
printf("%d\n",data[0][2]); //3

printf("%d",data[1][0]); //4
printf("%d",data[1][1]);  //5
printf("%d\n",data[1][2]);  //6

printf("%d",data[2][0]);//7
printf("%d",data[2][1]); //8
printf("%d",data[2][2]); //9
```

```c
// data[3][3] can be initialized like this also
int data[3][3]={{1,2,3},{4,5,6},{7,8,9}};
```

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

Initialing and Accessing a 2D Array: Example-2

Program

```c
int data[2][4] = {
{1,2,3,4}, //row 0 with 4 elements
{5,6,7,8}, //row 1 with 4 elements
    };
printf("%d",data[0][0]); //1
printf("%d",data[0][1]); //2
printf("%d",data[0][2]); //3
printf("%d\n",data[0][3]); //4

printf("%d",data[1][0]); //5
printf("%d",data[1][1]); //6
printf("%d",data[1][2]); //7
printf("%d",data[1][3]); //8
```

```c
// data[2][4] can be initialized like this also
int data[2][4]={{1,2,3,4},{5,6,7,8}};
```

	Col-0	Col-1	Col-2	Col-3
Row-0	1	2	3	4
Row-1	5	6	7	8

Ch: 4

Definition: String

- A String is a one-dimensional array of characters terminated by a `null('\0')`.

```
char name[10];
```

[0]	[1]	[2]	...	[9]

- Each character in the array occupies one byte of memory, and the last character must always be `null('\0')`.
- The termination character `('\0')` is important in a string to identify where the string ends.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
name[10]	D	A	R	S	H	A	N	\0		

Ch: 4

Declaring & Initializing String

Declaration

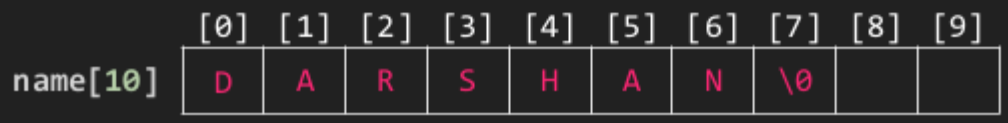
```
char name[10];
```

Initialization method 1:

```
char name[10]={'D','A','R','S','H','A','N','\0'};
```

Initialization method 2:

```
char name[10]="DARSHAN";
//'\0' will be automatically inserted at the end in this type of declaration.
```



1Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	What is Array?	Group of Elements having same name and type.
2	Array is_____datatype.	Derived
3	Array is used to represent _____	Collection
4	Types of array can be_____&	Single/One dimension& Multi/Two dimension

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	If array elements are initialized at the time of declaration then it is called _____ initialization.	Compile time
2	If array elements are initialized at the runtime then it is called_____initialization.	Runtime
3	In_____array initialization We have to assign fix value or size compulsory.	Compile time

4	Run time initialization of array is suitable for initializing_____array.	Large

Explain Array with matrix

Detail :-

- Array of Array is known as 2 – D array.
- 2-D array in C is also known as Matrix.
- Two dimensional array or multi-dimensional array are used to represent data in matrix form.
- Suppose , We declare array like a[3][3] then Matrix can be store as array like following:

Ch: 4

Syntax:

<data-type> <array_nm> [row] [column];

Example:

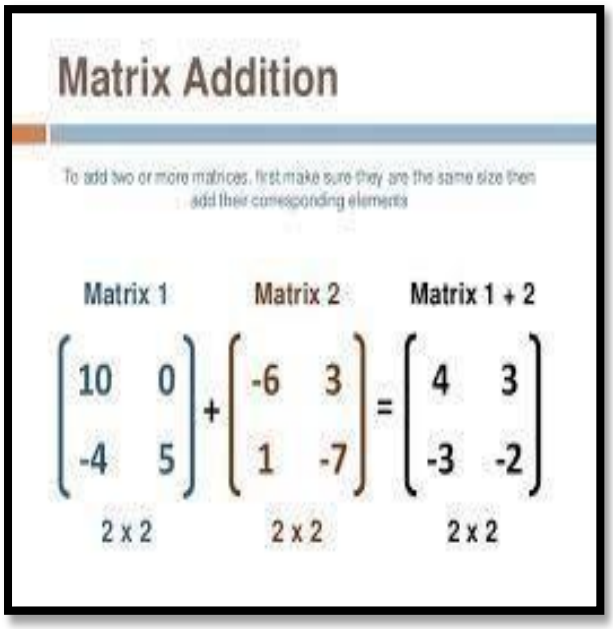
```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a[3][3],i, j;
        clrscr();
printf("\n\t Enter matrix of3*3 : ");
        for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        scanf("%d",&a[i][j]);  //read 3*3 array
                }
        }
  printf("\n\t Matrix is :\n");
                for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        printf("\t %d",a[i][j]);  //print 3*3 array
                }
          printf("\n");

            getch();
                }
```

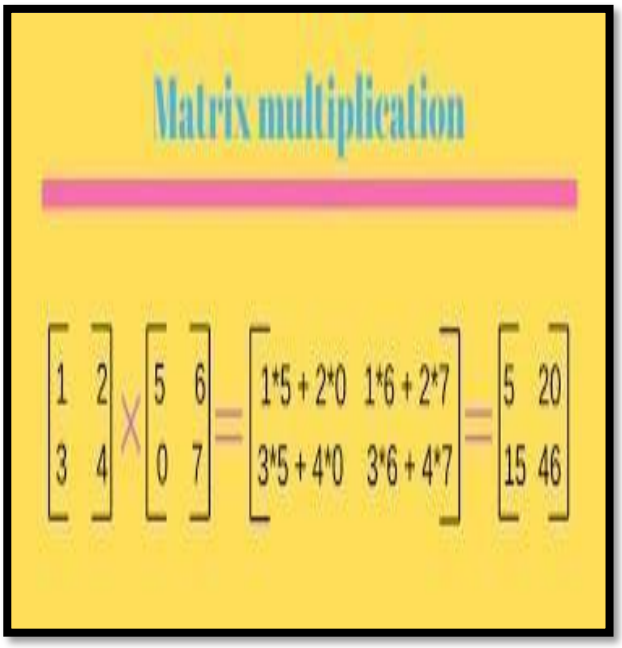
Ch: 4

*Enter matrix of 3*3 : 3 4 5 6 7 2 1 2 3*

Matrix is :

3	**4**	**5**
6	**7**	**2**
1	**2**	**3**





Program to Add Two Matrices

Ch: 4

```c
#include <stdio.h>
int main() {
  int r, c, a[100][100], b[100][100], sum[100][100], i, j;
  printf("Enter the number of rows (between 1 and 100): ");
  scanf("%d", &r);
  printf("Enter the 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 b%d%d: ", i + 1, j + 1);
      scanf("%d", &b[i][j]);
    }
```

Ch: 4

Output

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

// printing the result
printf("\nSum of two
matrices: \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");
```

```
Enter the number of rows (between 1 and
100): 2 Enter the 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
b11: -4
Enter element
b12: 5 Enter
element b13: 3
Enter element
b21: 5 Enter
element b22: 6
Enter element
b23: 3
```

Ch: 4

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Matrix can be represented by_____array.	2-D
2	In Array of Matrix elements can be represented by_____and_____.	Rows Columns

Explain String Array in brief?

The string can be defined as the one-dimensional array of characters terminated by a null ('\0'). The character array or the string is used to manipulate text such as word or sentences. Each character in the array occupies one byte of memory, and the last character must always be 0. The termination character ('\0') is important in a string since it is the only way to identify where the string ends. When we define a string as char s[10], the character s[10] is implicitly initialized with the null in the memory.

There are two ways to declare a string in c language.

1. By char array
2. By string

char ch[10]={'r', 'a', 'd', 'h', 'a', '\0'};

char ch[ ]="radha";

String Example in C.

```
#include<stdio.h>
#include<string.h>
int main()
{
 char ch[11]={'r','a','d','h','h', '\0'};
char ch2[11]="radha";
printf("Char Array Value is: %s\n", ch);
printf("String Literal Value is: %s\n", ch2);
```

```
return 0;

}
```

Ch: 4

Output:

Char array value is:radha

String literal value is:radha

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	String array is_____array of characters	2-D
2	String is array of_____object.	Char
3	Write down statement to declare string array	Char arr[3][12]
4	String array represent_____and _____.	Array size Total number of characters

Ch: 4