Arrays and Strings

Array:

→ A sequenced collection of related data items that shares of a common name.

→ we can use an array name salary to represent a set of salaries of a group of employees in an organization.

→ we can use arrays to represent not only simple lists of values but also table of data in two, three or more dimensions.

Types of Array:

1. One dimensional array
2. Two dimensional array
3. Multidimensional array.

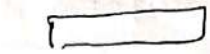
One dimensional array :-

→. A list of items can be given one variable name using only one subscript and such as a variable is called a single subscripted variable or one dimensional array.

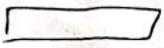
→ data type array_name [5]

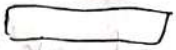
→ Int number [5].

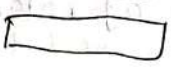
and the computer reserves Five Storage location.

number [0] 

" [1] 

" [2] 

" [3] 

" [4] 

→ The values to the array elements can be assigned as following.

number [0] = 32
number [1] = 40
number [2] = 22
number [3] = 56
number [4] = 73.

Declaration of one dimentional array:-

→ arays must be declared before they are used so that the compiler can allocate space for them in memory.

→ General form of array declaration is:

→ type Variable_name [size]

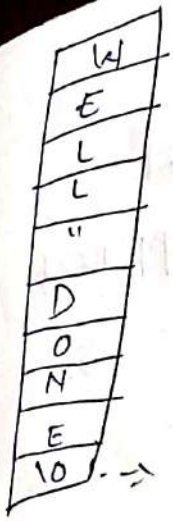
Int group [10]

float height [50]

Char name [10]

↳ Declare the name as a char. array (String) Variable that can hold a maximum of 10 characters.

"WELL DONE".



Initialization of one dimensional array:

→ After an array is declared its elements must be initialized. Otherwise they will contain garbage

→ An array can be initialized at either of following stages:

1. At compile time
2. At Run time

Compile time initialization:-

normal value assign ji pani

→ We can initialize the element of arrays in the same ways as the ordinary variables when they are declared.

type array_name [size] = {list of values};

for ex:

```
int number [3] = { 0, 5, 7 }
char name [] = { 'J', 'O', 'H', 'N', '\0' }
float average [4] = { 0.78, 0.56, 0.46, 0.38 }
```

Run time initialization:

An array can explicity initialized at run time. This approach is useallay applied for initializing large arrays:

```
for (i=0; i<100; i=i+1)
{
  if (i<50)
  sum[i]= 0.0;
  else
  sum[i]=1.0;
}
```

cout << "Enter array values:";

Ex:

```
#include <stdio.h>
#include <conio.h>
void main()
{ int array[10];
  int i;
  clrscr();
  Printf(" Enter the elements of an array :\n");
  for(i=0; i<10; i++)
  { scanf("%d", &array[i]);
  }
  Printf("The array elements are :\n");
  for(i=0; i<10; i++)
  Printf("array[%d]= %d \n", i, array[i]);
  getch()
  }
```

O/P

Enter the elements of an array:

10
20
30
40
50
60
70
80
90
100.

The array elements are:

array[0] = 10
array[1] = 20
array[2] = 30
array[3] = 40
array[4] = 50
array[5] = 60
array[6] = 70
array[7] = 80
array[8] = 90
array[9] = 100.

Two-dimensional array:-

The 2D array is organized as matrix which can be represented as the collection of rows and columns.

Syntax: data_type array_name [rows] [columns]

→ int two dimen [4][2];

Here 4 is the number of rows, and 2 is the no. of columns.