Array and String

Array:

An array is defined as collection of Same Type (homogenous) of data, stored in contiguous memory locations. You can store group of data of the same data type in an array.

Array is a contiguous memory locations. The lowest address corresponds to the first element and the highest address indicates to the last elements. Capacity of array indicates the number of elements of the given type in an array.

Properties:

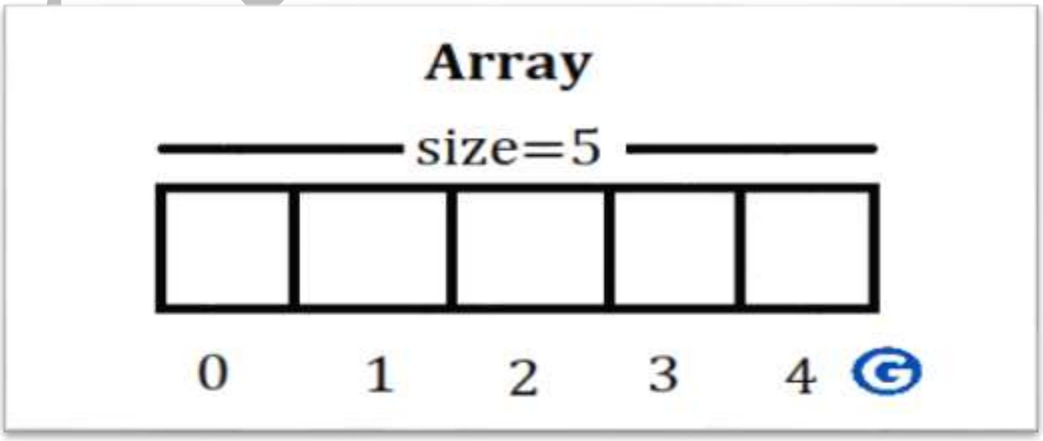
- Every element has a same data type and carries the same size (int 4 bytes)
- Elements of the array are stored at contiguous memory locations where the first element is stored at the smallest memory location.

Advantage:

- Code Optimization
- Random Access
- Ease of sorting

Disadvantage

- size fixed (Example: int 4 bytes).



GYANSABHA

One Dimensional Array:

An array with single subscript is known as one dimensional array. We can think of a one-dimensional array as a row, where elements are stored one after another. Consider array of six elements. Contiguous memory locations are used to store array elements in memory.

0	1	2	3	4	5

- **Declaration of One Dimensional Array:** An array must be declared before it appears in the program using a data type. At the same time, size of the array must be specified. This allows the compiler to decide on how much memory is to be reserved for the given array.

  Syntax:

  data_type array_name [size];

  - **data_type:** Data type of the elements stored in array.
  - **array_name:** Name of Array
  - **size:** Number of elements in the array

  Note:

  - Array index always start at first location 0.
  - If the size of an array is n. To access the last element then [n-1] used this code.

- **Initializing:** Array can be initializing at the time of declaration.

  Syntax:

  data_type array_name[size]={list of value}

  Example:

int num[5]={1,2,3,4,5}

This declaration initializes the first elements of an array num by 5 integers.

1	2	3	4	5
[0]	[1]	[2]	[3]	[5]

Note:

When the size of array is not specified, the allocate memory space is used to hold the initial value.

Example:

```
#include<stdio.h>
void main()
{
        int marks[5]={10, 20, 30, 40, 50};
        int i;
        printf("Marks in array:");
        for(i=0;i<=5;i++)
        {
                printf("%d\n", marks[i]);
        }
        getch();
}
```

Output:

```
10, 20, 30, 40, 50
```

- **Accessing and Displaying One Dimensional Array Elements:**

  Array must be declared before it is used. The individual data item of an array can be accessed using the name of the array and the subscript. The elements of an array can b accessed by specifying array name followed by subscript or index inside square brackets. Array subscript or index starts at 0.

  If the size of an array is 100 then the index start with 0(first element of index) while the last element is at the index 99. The first valid subscript is lower bound and last valid subscript is upper bound.

  Syntax:

  ```
  array_name[subscript]
  ```

  Example:

  ```
  course[10] /*consider the array 10 courses.*/
  ```

Two Dimensional Array:

Two dimensional array is organized as matrices they can represented as the collection of rows and columns.

- **Declaration of two dimensional array:** Declaration of two dimensional array as given below

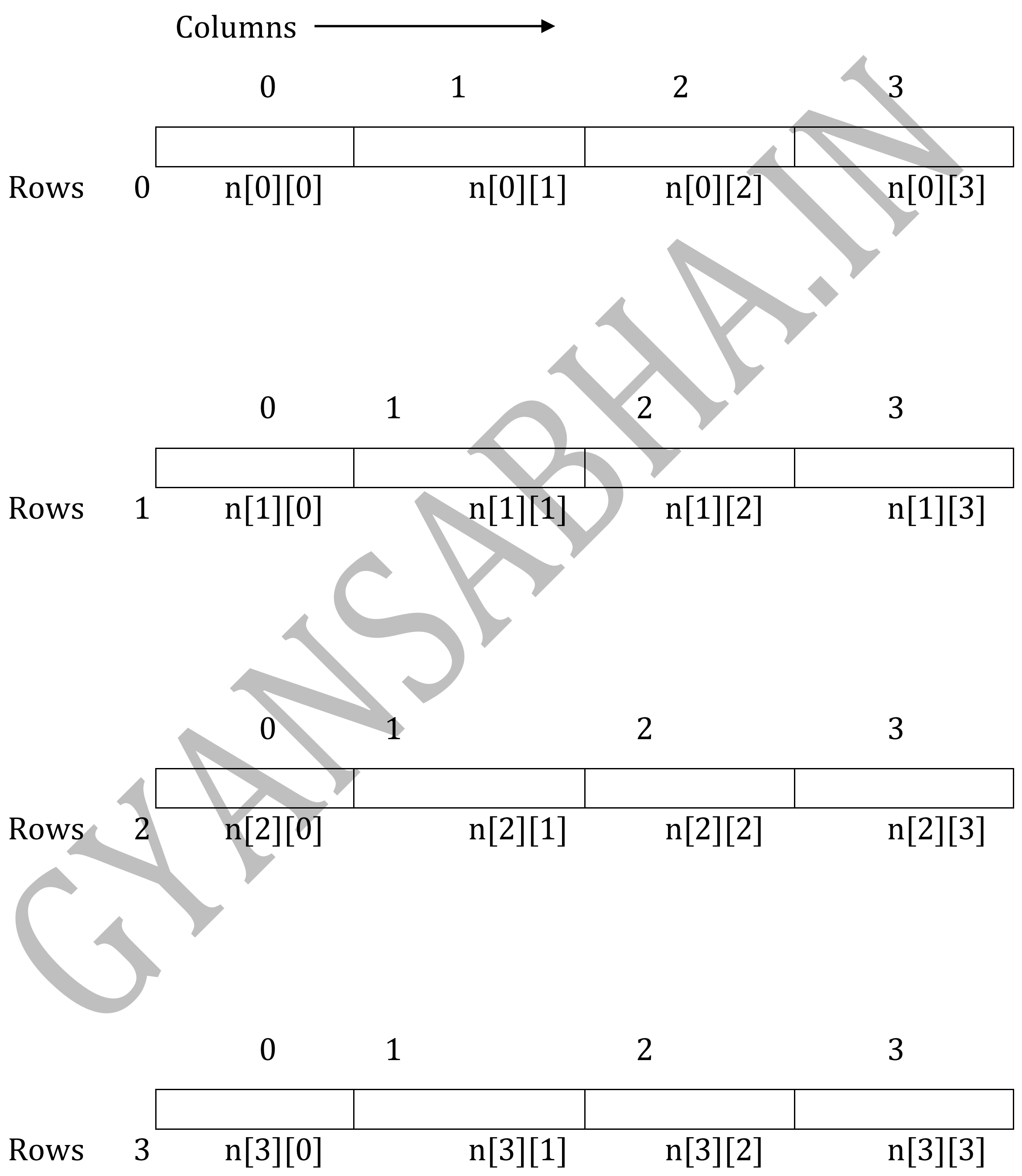
  Syntax:

  ```
  data_type array_name[rows][columns];
  ```

Example:

int n[4,4];

Array **n** has 4 row and 4 columns. The elements of array **n** are n[0][0], n[0][1], n[0][2],…… ,n[4][4]



- **Initialization of Two Dimensional Array:** Two dimensional array can be accessed using subscript. Your access the elements then **for loop** are required (1-row and 1-columns).

Example:

```
#include <stdio.h>
void main()
{
    int n[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, j;
    printf("Elements in Matrix:\n");
    for(i=0;i<3;i++)
    {
            for(j=0;j<3;j++)
            {
                    printf("%d", n[i][j]);
            }
    printf("\n");
    }
}
```

Output:

```
123
456
789
```

○ Accessing and Displaying Two Dimensional Array Elements:

To access and display elements in two dimensional array. for loop are used. One for loop is used to access elements on matrix and second for loop is used to display elements in matrix. it is better programming practice to enclose each row within a separate subset of curly braces, to make the program more readable.

Array and Function:

In this type of function call, the actual parameter is copied to the formal parameters. While passing arrays as arguments to the function, only the name of the array is passed.

Syntax:

```
function_name(array_name);
```

Example:

```
#include<stdio.h>
float average(float age[]);
void main()
{
        float avg, age[] = {1, 2, 3, 4, 5};
        avg = average(age);
        printf("Average age=%f", avg);
}
float average(float age[])
{
        int i;
        float avg, sum=0.0;
        for(i=0;i<6;++i)
        {
                sum += age[i];
        }
        avg = (sum/5);
        return avg;
}
```

Output:

```
average age=3
```