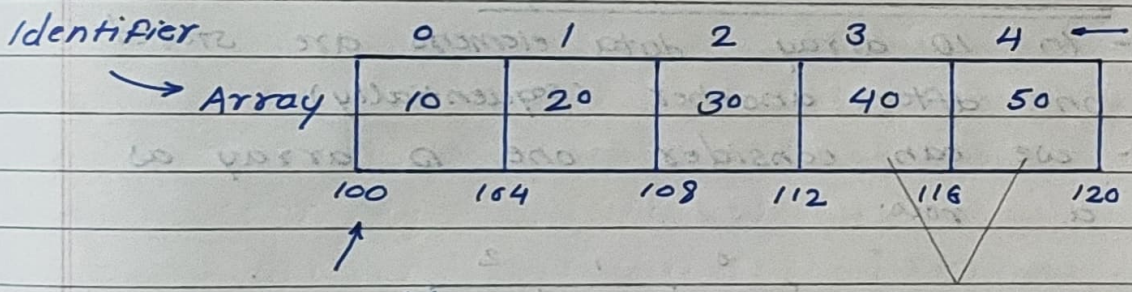
Assignment No. - 7

1. What is mean by array ? Explain the use of array with example.

- Array is the collection of homogeneous data elements in it.
- Array is the derived data type & it is a linear datastructure.
- Array stores the data in the indexed format in it.
- Array stores the data elements in an contiguous memory locations.

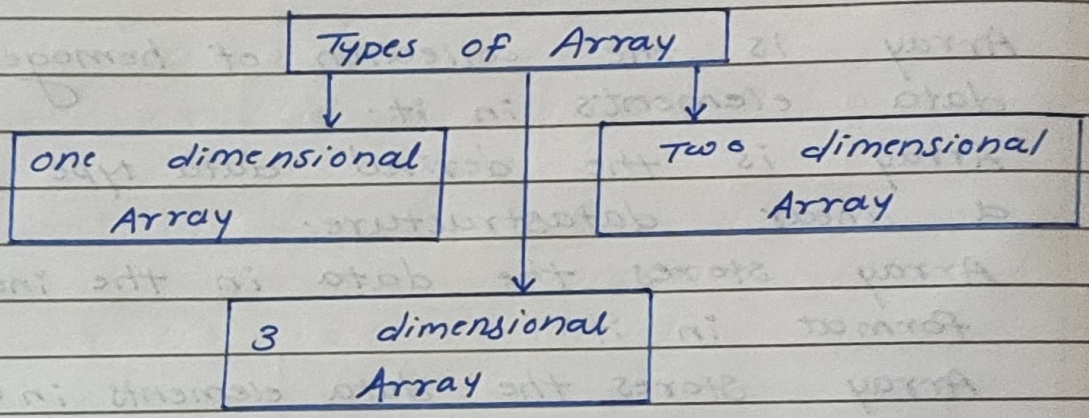


- Array is used in mathematical problems like matrics etc.
- Array is also used to store tree, graph, stack etc.
- eg - If we want to create a / store 5 integers we can use array.

i.e int arr [5] = {10, 20, 30, 40, 50}.

2. What is the types of array?

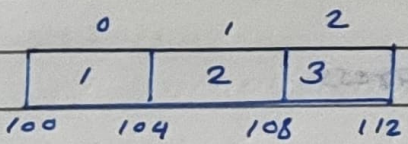
- According to dimensions array is of 3 types.



- <u>one dimensional Array</u>

- In 1D array, data elements are stored one after another (sequentially).
- we can consider one D array as a row.

e.g Arr



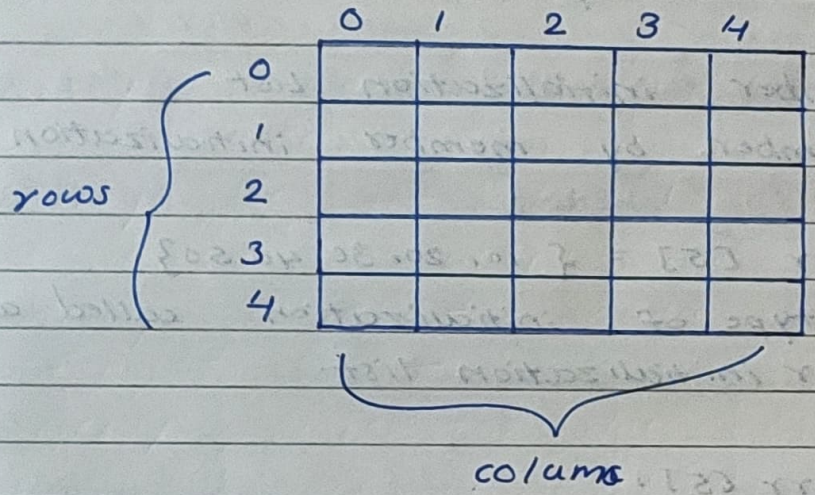
int Arr[3] = {1, 2, 3};

- Syntax :

Data type Array_name [size];

- **Two dimensional Array**
  - The data stored in 2D array in row & columns format. (also called as matrix).

e.g int arr [5] [5];

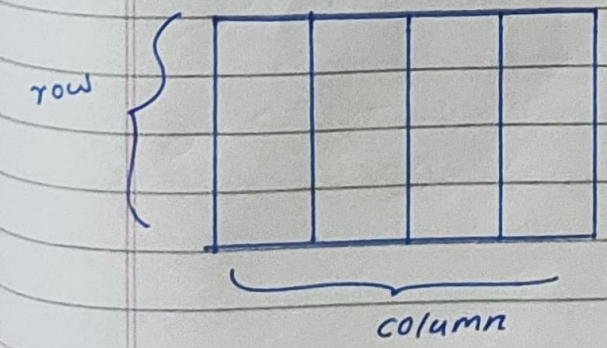


- Syntax : data type Array_name [size of 1D array] [size of 2D array]

- **Three dimensional Array**
  - 3D array contains three dimensions, so it can be considered as an array of 2D arrays.

arr [2] [row] [column]

arr [0] [row] [col]



3. what are the ways in which we can initialize the array elements.

- data elements are initialized in two ways in array -

1. member initialization List
2. member by member initialization

1. int arr [5] = { 10, 20, 30, 40, 50}

- such type of initialization called as member initialization list.

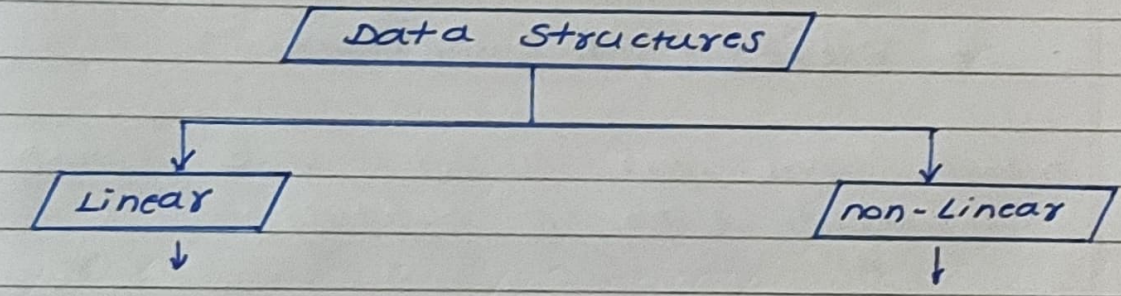
2.
```
int arr [5];
int arr [0] = 1;
int arr [1] = 2;
int arr [2] = 3;
int arr [3] = 4;
int arr [4] = 5;
```

- such type of initialization called member by member initialization.

4. what is meant by data structure? and what are the types of data structures

- Data structure :
  It is about the arranging data in perticular manner so, that it can be easily accessed and stores efficiently.

- There are two types of data structures



Data is stored in the linear format or we can say in sequential manner.

eg. array, stack, queue, linked list, ..

Data is stored in the non-linear format

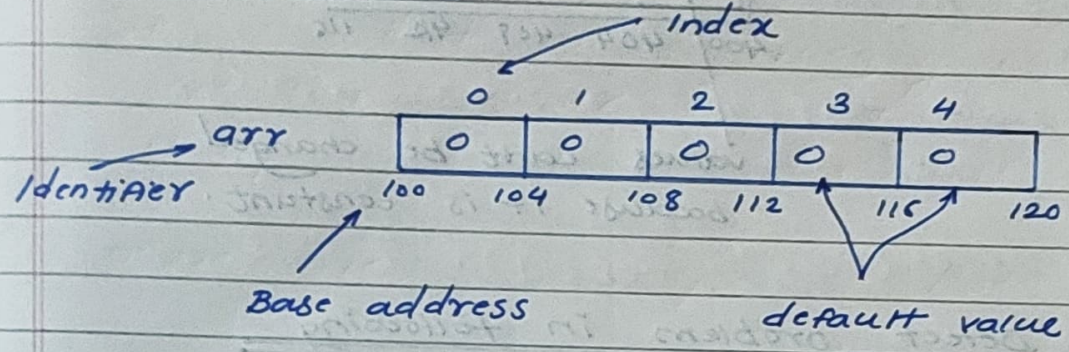
e.g graph, Tree, Hash, map..etc.

5. what are the ways in which we can allocate memory for array?

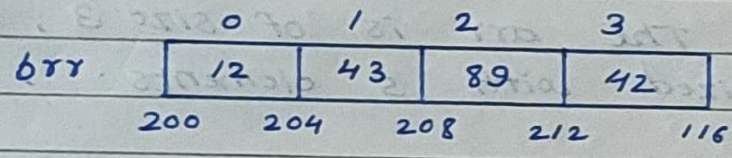
- The memory for array can be initialized or allocate in two ways i.e
  1. static memory allocation (compile time)
  2. Dynamic memory allocation (run time)

6. Read the below and draw diagramatic representation.

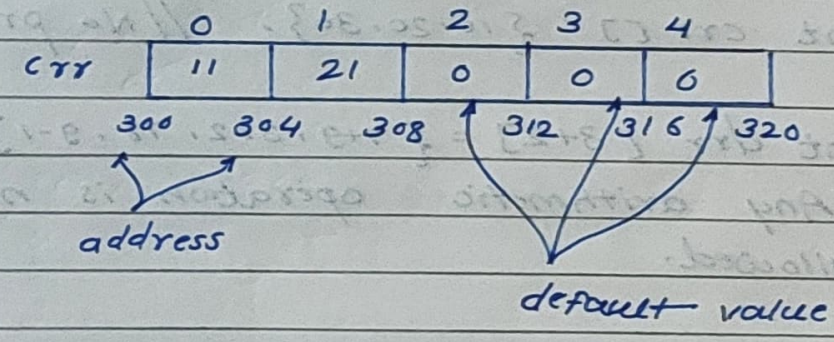
1. int arr [5];



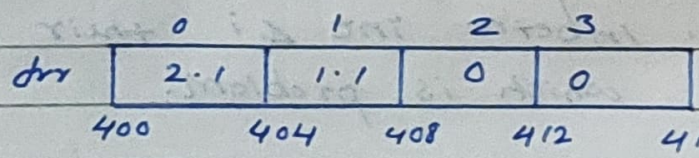
2. int brr [4] = {12, 43, 89, 42};



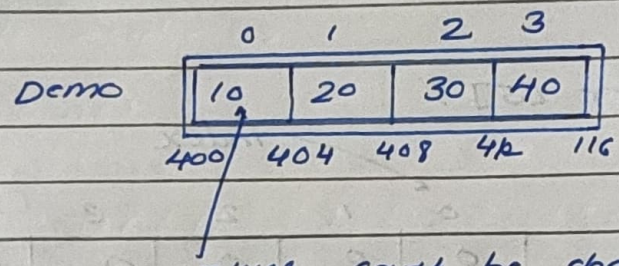
3. int crr [5] = {11, 21};



4. float drr [4] = {2.1, 1.1}



5. const int Demo [4] = {10, 20, 30, 40};



7. Detect problem in following

1. int arr[3] = {12, 23, 45, 65, 68};

- Here, The arr is of size 3, which is initialized with 5 elements.

2. int brr[];
- Array size is not declared.

3. int crr[] = {10, 20, 30}; // No problem

4. int drr [3+2] = {7+9, 3*2, 78, 9-1};
- Any arithmetic operation is not allowed.

5. int i = 4;
int arr [i] = {23, 6, 89, 32};

- Here, inbetn. int & i their should be space which is problem.

8. what is meant by size of operator & explain with example.

- **Size of operator** is used to find the size of variable i.e how memory it holds

- eg.

```
int no = 10;
int Arr[2] = {10,20};

printf("%d", sizeof(no));      // 4 byte
printf("%d", sizeof(Arr));     // 12 byte
```

- Syntax :

```
sizeof( variable name which is declared already );
```

9. **Predict the output :**

```
#includ <stdio.h>
int main()
{
   int arr[3] = {21,43,54};
   int x  = 0;

x =  arr[2] + 21 + arr[0] ;   // 54+21+21 = 96
   x++ ;          // 96+1  = 97
printf("%d", x);

   return 0;
}
```

o/p → 97

10. predict the output:

```
# include <stdio.h>
int main()
{
    float arr[4] = {98.3, 4.3, 51.6, 7.6};
    int i = 0;

    printf("%f", arr[i]); // 98.3
    i++;

    printf("%f", arr[i]); // 4.3

    printf("%f", arr[i]); // 51.6

    return 0;
}
```

o/p → 98.3 4.3 51.6