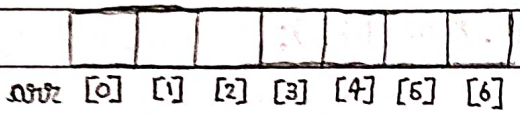
PAGE NO.:

Arrays

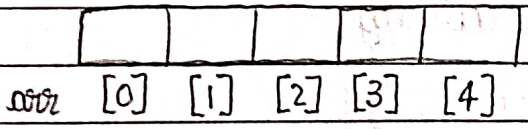
- Array : set of pairs (a value and a index)
- An Array is a finite collection of similar elements stored in adjacent memory location.
- Array is represented in memory as consective memory blocks.



Implementation of Array

Declaration : type arr name [size]

Example : int arr [5]



arr [0] = Base address (Random address assigned to arr)
arr [k] = Base address + k * size of (int)

Implementation of Array

```
#include <stdio.h>
int main () {
    int arr[5];
    int i;
    for (i=0; i<5; i++) {
        scanf("%d", &arr[i]);
    }
    for (i=0; i<5; i++) {
        printf("%d", arr[i]);
    }
    return 0;
}
```

Address of Array

```
for (i=0; i<5; i++) {
    printf("%p\n", &arr[i]);
}
```

PAGE NO.:

Two - Dimensional Arrays

- A 2-dimensional arrays is a collection of elements placed in 'm' rows and 'n' colums.
- Two dimensional arrays are used for representing table and matrics.

Declaration

```
type arr name [ row size ][ col size ]
```

for example :

```
int arr[3][5]
```

Type - int
Name - arr
3 rows and 5 columns

Initialization

Directly Initialize while declaring

- `int arr[2][3] = { 1,5,3,8,7,4}`
- `arr [0][2] = 9`

Use braces to separate :

```
int arr[2][3] = {
                  { 1,5,3},
                  { 8,7,4},
                }
```

```c
#include <stdio.h>
#include <stdlib.h>

int main () {
    int arr[2][3];
    int i, j;
    int counter = 1;

    for ( i=0; i<2 ; i++ ) {
        for ( j=0 ; j<3 ; j++ ) {
            arr[i][j] = counter ;
            counter = counter+1 ;
        }
    }
    for ( i=0; i<2 ; i++ ) {
        for (j=0; j<3; j++ ) {
            printf ("%d", arr[i][j]);
        }
        printf ("\n");
    }
    return 0;
}
```