**DAY-9**

**ARRAYS:**

**//decalaration part**

dt ArrName[CAP];

int arr[10];

float arr1[10];

struct Emp

{

int id;

char name[20];

};

struct Emp e[10];

**// access elements of arr to store or evealute**

ArrName[indexValue] = Value;

=> indexValue from 0 to CAP-1

arr[2] = 20;

int a[2] = {1,2};

dt arrName[ROW][COL];

int a[2][3] = {{1,2,3},{4,5,6}};

int a1[2][3] = {1,2,3,4,5,6};

1 2 3

4 5 6

2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012

1 2 3 4 5 6

a[0][0] a[0][1] a[0][2] a[1][0] a[1][1] a[1][2]

printf("\n%d",a[1][2])

🡪Arrays are of 4 Types:

1. static array

2. dynamic array

3. stretchable array

4. mutable array

1**. static array:** the size of array is known before to the compl time.

ex:

int arr[5];

2**. dynamic arr:** size of the array is allocated/known at run time.

malloc, calloc, realloc => stdlib.h

3**. stretchable array:**

size of the array are increased or dec depending on the need

for dy. array

malloc, calloc, realloc

**4. mutable:**

the size of the array is known or allocted at the time of linking and before execution.

**FUNCTIONS:**

1. std lib function

printf, sqrt, abs,pow

2. user defined

user is defining his/her own task to be performed

rdt fName(input args)

{

sts;

return rdt;

}

int add(int,int);

int add(int val1, int val2)

{

float result = val1+val2;

return result;

}

**Code:**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

// functions without input args without return dt

void disp();

//functions with input args and return dt

int add(int, int);

//functions with input args and no return dt

void changeName(char \*);

int main()

{

char Name[20]="Bhima";

disp();

printf("\nAddition: %d\n",add(10,30));

printf("\nName before change: %s",Name);

changeName(Name);

printf("\nName after change: %s",Name);

printf("\n\n");

return EXIT\_SUCCESS;

}

void disp()

{

int i;

for(i=0;i<30;i++)

printf("=");

printf("\n=========Welcome=======\n");

for(i=0;i<30;i++)

printf("=");

printf("\n\n");

}

int add(int v1, int v2)

{

return (v1+v2);

}

void changeName(char \*name)

{

strcpy(name,"TestName");

}