15-06-2024

FUNCTION RETURNING A POINTER =>

```
-> int* add(int, int);
main()
{
    int a=5, b=6, *c;
    c = add(a, b);
    printf("%d", *c);
}
int* add(int a, int b)
{
    int c;
    c = a+b;
    return &c;
}
```

POINTER TO A FUNCTION =>

```
-> void add();
   main()
   {
     void (*P)();
     P = add;
     (*P)();
   }
```

```
int add(int, int);
main()
{
    int a=5, b=6, c;
    int (*p)(int, int);
    p = add;
    c = (*p)(a, b);
    printf("%d", c);
}
int add(int a, int b)
{
    return a+b;
}
```

```
int add(int, int)
int add(int *, int *)
int* add(int, int)
int* add(int *, int *)
int (*add)(int, int)
int (*add)(int*, int *)
int* (*add)(int, int)
int* (*add)(int *, int *)
```

GENERIC POINTER:- Any pointer is converted to void pointer & void pointer is converted to any other pointer.

∴ Void pointer is said to be Generic Pointer.

EX:-

```
-> main()
{
    int a=5;
    void *p;
    int *q;
    p= &a;
    q= (int *)p;
    printf("%d", *q);
}
```

O/P:- 5

POINTER TO A POINTER:-

```
-> main()
{
    int a=5, *p, **q, ***r;
    p= &a;
    q= &p;
    r= &q;
    printf("%d %d %d %d", a, *p, **q, ***r);
}
```

O/P:-

5 5 5 5