C_102 => Callback Functions using Function Pointer.

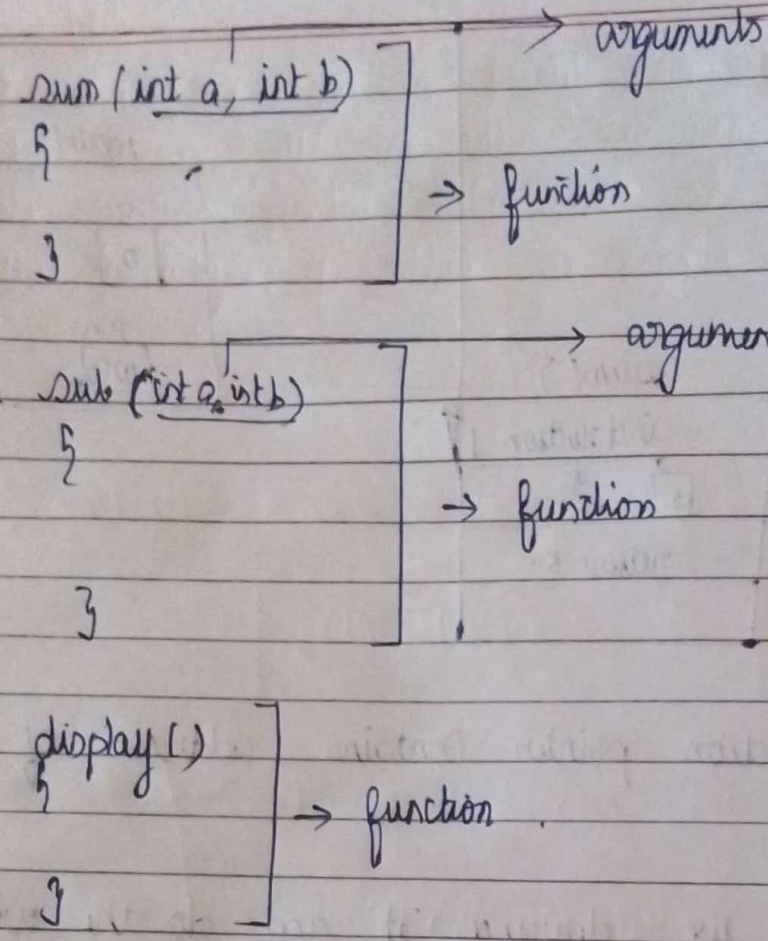
Why we use function Pointer :-

* It is used for callback functions.
* Then what is callback function?

Function Pointer :-

* It is like normal pointer variable but contains a address of a function code.

* But normal points contains address of value or data.



* In a function, instead of passing arguments we pass a ~~function~~ address of function which calls the spcific function.

* In one function we can call the other functions ~~multiple times~~ by passing address of other functions as arguments to the calling function.

Call back function → We are passing address of other functions to a function by this way we can call other functions from a one single function.

* To pass address of function pointer in a function; first we should declare function pointer.

Example Program;

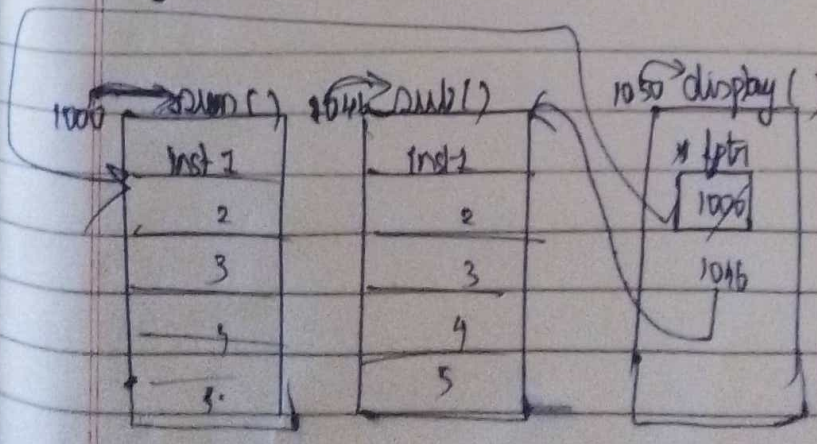
```
void sum(int a, int b)
{
  printf("%d", a+b);
}

void sub(int a, int b)
{
  printf("%d", a-b);
}

void display(void (*fptr)(int, int))
{
    *fptr(5,1);
}

void main()
{
  display(sum);   // 1000
  display(sub);   // 1046
}
```

→ In display function sum and sub are call back functions.



Date
Page

Note :-

* using one function pointer we are going to access all the other functions; so the function pointer declaration should match the prototype of other functions.

```c
#include <stdio.h>
#include <stdlib.h>
/** 1- Call Back Functions in C **/
void sum(int a, int b)
{
 int sum;
 sum= a+b;
 printf("Sum is: %d\n",sum);
}
void sub(int a, int b)
{
 int sub;
 sub= a-b;
 printf("Sub is: %d\n",sub);
}
void display(void (*ptr)(int,int))
 {
  (*ptr)(6,3);
 }

int main()
{
 display(sum);
 display(sub);
}
```