

HOW TO CALL C FUNCTIONS IN A PROGRAM?

There are two ways that a C function can be called from a program. They are,

- Call by value
- Call by reference

CALL BY VALUE:

- In call by value method, the value of the variable is passed to the function as parameter.
- The value of the actual parameter can not be modified by formal parameter.
- Different Memory is allocated for both actual and formal parameters. Because, value of actual parameter is copied to formal parameter.

EXAMPLE PROGRAM FOR C FUNCTION (USING CALL BY VALUE):

- In this program, the values of the variables “m” and “n” are passed to the function “swap”.
- These values are copied to formal parameters “a” and “b” in swap function and used.

```
1 #include<stdio.h>
2 // function prototype, also called function declaration
3 void swap(int a,
4 int b);
5 int main()
6 {
7     int m = 22, n = 44;
8     // calling swap function by value
9     printf(" values before swap m = %d \nand n = %d", m, n);
10    swap(m, n);
11 }
12
13 void swap(int a, int b)
14 {
15     int tmp;
16     tmp = a;
17     a = b;
18     b = tmp;
19     printf(" \nvalues after swap m = %d\n and n = %d", a, b);
```

OUTPUT:

values	befo	swa	m	=	22
and	n		=		44
values	re	p	m	=	44
and n =					
22	after	swa			
		p			

2. CALL BY REFERENCE:

- In call by reference method, the address of the variable is passed to the function as parameter.
- The value of the actual parameter can be modified by formal parameter.
- Same memory is used for both actual and formal parameters since only address is used by both parameters.

EXAMPLE PROGRAM FOR C FUNCTION (USING CALL BY REFERENCE):

- In this program, the address of the variables “m” and “n” are passed to the function “swap”.
- These values are not copied to formal parameters “a” and “b” in swap function.
- Because, they are just holding the address of those variables.
- This address is used to access and change the values of the variables.

```

1 #include<stdio.h>
2 // function prototype, also called function declaration
3 void swap(int *a,
4 int *b);
5 int main()
6 {
7     int m = 22, n = 44;
8     // calling swap function by reference
9     printf("values before swap m = %d \n and n = %d",m,n);
10    swap(&m, &n);
11 }
12
13 void swap(int *a, int *b)
14 {

```

```

15  int tmp;
16  tmp =
17  *a;
18  *a = *b;
19  *b = tmp;
20  printf("\n values after swap a = %d \nand b = %d", *a, *b);
}

```

COMPILE
& RUN

OUTPUT:

values before swap				
m = 22 and	n	=		44
values	swap	a	=	44
	aft			
er and b = 22				

Difference between Call by Value and Call by Reference

Functions can be invoked in two ways: **Call by Value** or **Call by Reference**. These two ways are generally differentiated by the type of values passed to them as parameters.

The parameters passed to function are called *actual parameters* whereas the parameters received by function are called *formal parameters*.

Call By Value: In this parameter passing method, values of actual parameters are copied to function's formal parameters and the two types of parameters are stored in different memory locations. So any changes made inside functions are not reflected in actual parameters of caller.

Call by Reference: Both the actual and formal parameters refer to same locations, so any changes made inside the function are actually reflected in actual parameters of caller.

CALL BY VALUE

CALL BY REFERENCE

While calling a function, instead of passing the values of

While calling a function, we pass values of variables to it. Such functions are known as “Call By Values”.

variables, we pass address of variables(location of variables) to the function known as “Call By References.

In this method, the value of each variable in calling function is copied into corresponding dummy variables of the called function.

In this method, the address of actual variables in the calling function are copied into the dummy variables of the called function.

With this method, the changes made to the dummy variables in the called function have no effect on the values of actual variables in the calling function.

With this method, using addresses we would have an access to the actual variables and hence we would be able to manipulate them.

In call by values we cannot alter the values of actual variables through function calls.

In call by reference we can alter the values of variables through function calls.