

POINTERS AS FUNCTION ARGUMENTS:

A pointer can be used as an argument in function declaration. When a function with a pointer argument is called, the calling program will pass the address(not the value) of a variable to the argument.

In C language, a function can be called by the calling program in two ways:

- 1. Call by value.*
- 2. Call by reference.*

Call by Value:

When a function is called by the calling program, the values to the arguments in the function are supplied by the calling program. The values supplied can be used inside the function. Any alteration to the

value inside the function is not accepted in the calling program but the change is locally available in the function. This method is referred as calling a function by value.

Call by Reference:

A function can be declared with pointers as its arguments. Such functions are called by the calling program with the address of a variable as argument from it. The addresses of the variables are substituted to the pointers and any alteration to its value inside the function is automatically carried out in that location. The change is indirectly made and is accepted by the calling program.

This method is referred as calling a function by reference.

Comparison between Call by Value and Call by Reference:

Call by Value	Call by Reference
<i>1.This is the usual method to call a function in which only the value of the variable is passed as an argument.</i>	<i>1. This method, the address of the variable is passed as an argument.</i>
<i>2.An alteration in the value of the argument passed is local to the</i>	<i>2.Any alteration in the value of the argument</i>

<i>function and is not accepted in the calling program.</i>	<i>passed is accepted in the calling program.</i>
<i>3.Memory location occupied by formal and actual arguments is different.</i>	<i>3.Memory location occupied by formal and actual arguments is same and there is a saving of money.</i>
<i>4.Since a new location is created, this method is slow.</i>	<i>4.Since the existing memory location is used through its address, this method is fast.</i>
<i>5.There is no possibility of wrong data manipulation since the arguments are directly used in an expression.</i>	<i>5.There is a possibility of wrong data manipulation since the addresses are used in an expression.</i>

