**Call by values-**

In the Call by Value method, the called function creates new variables to store the value of the arguments passed to it. Therefore, the called function uses a copy of the actual arguments to perform its intended task.

If the called function is supposed to modify the value of the parameters passed to it, then the change will be reflected only in the called function. In the calling function no change will be made to the value of the variables.

#include<stdio.h>

void swap(int, int);

void main()

{

int a, b;

printf(“Enter the values”);

scanf(“%d % d”, &a, &b); // a= 10 b = 20

printf(“Values of a = %d and b = %d before swapping”, a, b); // 10 20

swap(a, b); // call by value

printf(“Values of a = %d and b = %d after swapping”, a, b); // 10 20

}

void swap(int a, int b){

int temp;

temp = a;

a= b;

b=temp;

printf(“Values of a = %d and b = %d is ”, a, b); // 20 10

}

#include<stdio.h>

void add( int n);

int main()

{

int num = 2;

add(num);

printf("\n The value of num after calling the function = %d", num);

return 0;

}

void add(int n)

{

n = n + 10;

printf("\n The value of num in the called function = %d", n);

}

void recurse()

{

... .. ...

recurse();

... .. ...

}