

A function can call itself. Such a process is called "recursion".

In C, it is possible for the functions to call themselves. A function is called 'recursive' if the statement within the body of a function calls the same function. Some time called "circular definition".

Example

function without recursion ↓

```

main ()
{
  int a, fact;
  printf("In Enter any no.");
  scanf("%d", &a);
  fact = factorial(a);
  printf("Factorial value = %d", fact);
}

```

Simple function calling →

Definition of function →

```

factorial (int x)
{
  int f = 1, i;
  for (i = x; i >= 1; i--)
    f = f * i;
  return (f);
}

```

function with recursion ↓

```

main ()
{
  int a, fact;
  printf("In Enter any no.");
  scanf("%d", &a);
  fact = rec(a);
  printf("Factorial value = %d", fact);
}

```

```

rec (int x)
{
  int f;
  if (x == 1)
    return (1);
  else
    f = x * rec(x-1);
  return (f);
}

```

in definition of function rec (int x) we use f = x \* rec(x-1) again, means function call itself

recursion

p-2 Q. W.a.p to generate the Fibonacci Series for a given number using recursive function.

```
#include <stdio.h>
int fibonacci (int)
{
    if (i == 0)
    {
        return 0;
    }
    if (i == 1)
    {
        return 1;
    }
    return fibonacci (i-1) + fibonacci (i-2);
}

int main ()
{
    int i;
    for (i=0; i < 10; i++)
    {
        printf ("%d \n", fibonacci (i));
    }
    return 0;
}
```

function call itself  
i.e. recursion



3

output

0  
1  
1  
2  
3  
5  
8  
13 21 34