

# C - RECURSION

[http://www.tutorialspoint.com/cprogramming/c\\_recursion.htm](http://www.tutorialspoint.com/cprogramming/c_recursion.htm)

Copyright © tutorialspoint.com

Recursion is the process of repeating items in a self-similar way. Same applies in programming languages as well where if a programming language allows you to call a function inside the same function that is called recursive call of the function as follows.

```
void recursion()
{
    recursion(); /* function calls itself */
}

int main()
{
    recursion();
}
```

The C programming language supports recursion, i.e., a function to call itself. But while using recursion, programmers need to be careful to define an exit condition from the function, otherwise it will go in infinite loop.

Recursive functions are very useful to solve many mathematical problems like to calculate factorial of a number, generating Fibonacci series, etc.

## Number Factorial

Following is an example, which calculates factorial for a given number using a recursive function:

```
#include <stdio.h>

int factorial(unsigned int i)
{
    if(i <= 1)
    {
        return 1;
    }
    return i * factorial(i - 1);
}

int main()
{
    int i = 15;
    printf("Factorial of %d is %d\n", i, factorial(i));
    return 0;
}
```

When the above code is compiled and executed, it produces the following result:

```
Factorial of 15 is 2004310016
```

## Fibonacci Series

Following is another example, which generates Fibonacci series for a given number using a recursive function:

```
#include <stdio.h>

int fibonacci(int i)
{
    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\t", fibonaci(i));  
    }  
    return 0;  
}
```

When the above code is compiled and executed, it produces the following result:

```
0 1 1 2 3 5 8 13 21 34
```