

C-RECURSION

Recursion is the process of repeating items in a self-similar way. In programming languages, if a program allows you to call a function inside the same function, then it is called a recursive call of the function.

```
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 into an infinite loop.

Recursive functions are very useful to solve many mathematical problems, such as calculating the factorial of a number, generating Fibonacci series, etc.

Number Factorial

The following example calculates the factorial of a given number using a recursive function –

[Live Demo](#)

```
#include <stdio.h>

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

int main() {
    int i = 12;
    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 12 is 479001600
```

Fibonacci Series

The following example generates the Fibonacci series for a given number using a recursive function –

[Live Demo](#)

```
#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\t\n", fibonacci(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
```