

Recursion in function

C is a powerful programming language having capabilities like an iteration of a set of statements 'n' number of times. The same concepts can be done using functions also. In this chapter, you will be learning about recursion concept and how it can be used in the C program.

What is Recursion

Recursion can be defined as the technique of replicating or doing again an activity in a self-similar way calling itself again and again, and the process continues till specific condition reaches. In the world of programming, when your program lets you call that specific function from inside that function, then this concept of calling the function from itself can be termed as recursion, and the function in which makes this possible is called recursive function.

Here's an example of how recursion works in a program:

Example Syntax:

```
void rec_prog(void) {  
    rec_prog(); /* function calls itself */  
  
int main(void) {  
    rec_prog();  
    return 0;  
}
```

C program allows you to do such calling of function within another function, i.e., recursion. But when you implement this recursion concept, you have to be cautious in defining an exit or terminating condition from this recursive function, or else it will continue to an infinite loop, so make sure that the condition is set within your program.

Factorial Program using Recursion

Example:

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
int fact(int f) {  
    if (f & lt; = 1) {  
        printf("Calculated Factorial");  
        return 1;  
    }  
    return f * fact(f - 1);  
}
```

```
int main(void) {  
    int f = 12;  
    clrscr();  
    printf("The factorial of %d is %d \n", f, fact(f));  
    getch();  
    return 0;  
}
```

Fibonacci Program using Recursion

Example:

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
int fibo(int g) {
```

```
    if (g == 0) {
```

```
        return 0;
```

```
    }
```

```
    if (g == 1) {
```

```
        return 1;
```

```
    }
```

```
    return fibo(g - 1) + fibo(g - 2);
```

```
}
```

```
int main(void) {
```

```
    int g;
```

```
    clrscr();
```

```
    for (g = 0; g < 10; g++) {
```

```
        printf("\nNumbers are: %d \t ", fibonacci(g));
```

```
    }
```

```
    getch();
```

```
    return 0;
```

```
}
```

