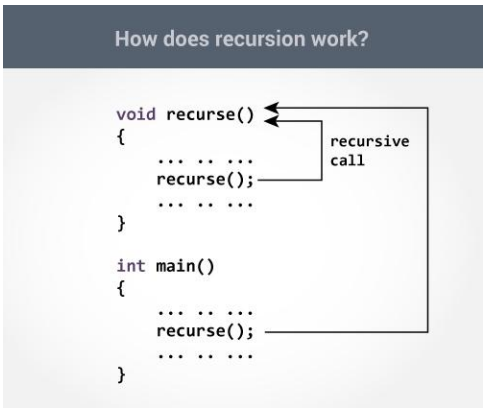
RECURSIVE FUNCTIONS

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. Recursion is used to solve various mathematical problems by dividing it into smaller problems. In programming , it is used to divide complex problem into simpler ones and solving them individually.



But while using recursion, we need to be careful to define an exit condition from the function, otherwise it will go into an infinite loop.

An example:

```c
#include <stdio.h>
int sum(int n);
int main()
{
    int number, result;
    printf("Enter a positive integer: ");
    scanf("%d", &number);
    result = sum(number);
    printf("sum=%d", result);
}
```

```
int sum(int num)
{
    if (num!=0)
        return num + sum(num-1); // sum() function calls itself
    else
        return num;
}
```

Initially, the sum() is called from the main() function with number passed as an argument. Suppose, the value of num is 3 initially. During next function call, 2 is passed to the sum() function. This process continues until num is equal to 0. When num is equal to 0, the if condition fails and the else part is executed returning the sum of integers to the main() function.

Types of recursion:

1) Direct recursion

2) Indirect recursion

Direct Recursion

A function is said to be direct recursive if it calls itself directly.

Indirect Recursion

A function is said to be indirect recursive if it calls another function and this new function calls first calling function again.

Advantages and Disadvantages of Recursion

Recursion makes program elegant and cleaner. All algorithms can be defined recursively which makes it easier to visualize and prove.

If the speed of the program is vital then, you should avoid using recursion. Recursions use more memory and are generally slow. Instead, you can use loop.