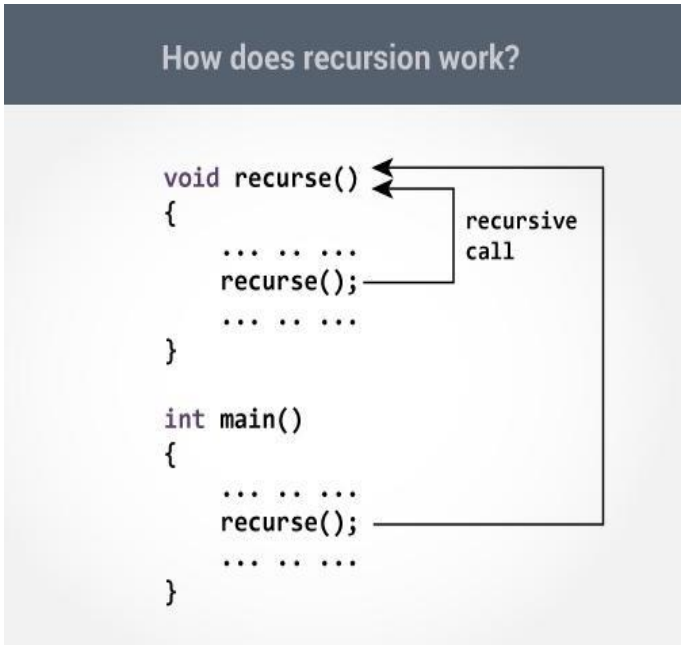
RECURSIVE FUNCTIONS

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.

A function is said to be 'recursive' if a statement within the body of a function calls the same function. Sometimes it is also called as **"circular definition"**.



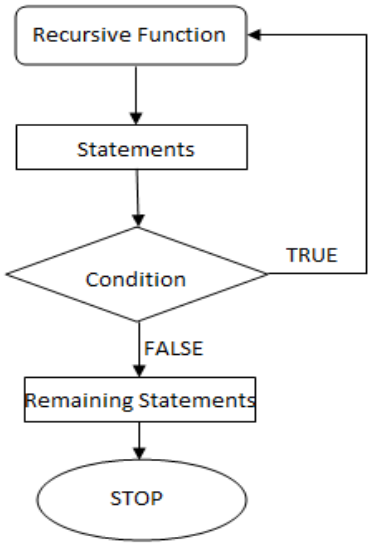


Fig: Flowchart showing recursion

- An Example: **TO FIND SUM OF NATURAL NUMBERS USING RECURSION**.
  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 as shown in the below example.

```
#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;
}
```

Processor uses stack to manage these recursive functions which involves push and pop operations for storing and retrieving the local variables and return addresses respectively, and cleans up the stack upon encountering the return statement, and this continues….

Processor uses stack to manage procedures and function calls

As shown below :