RECURSION FUNCTION

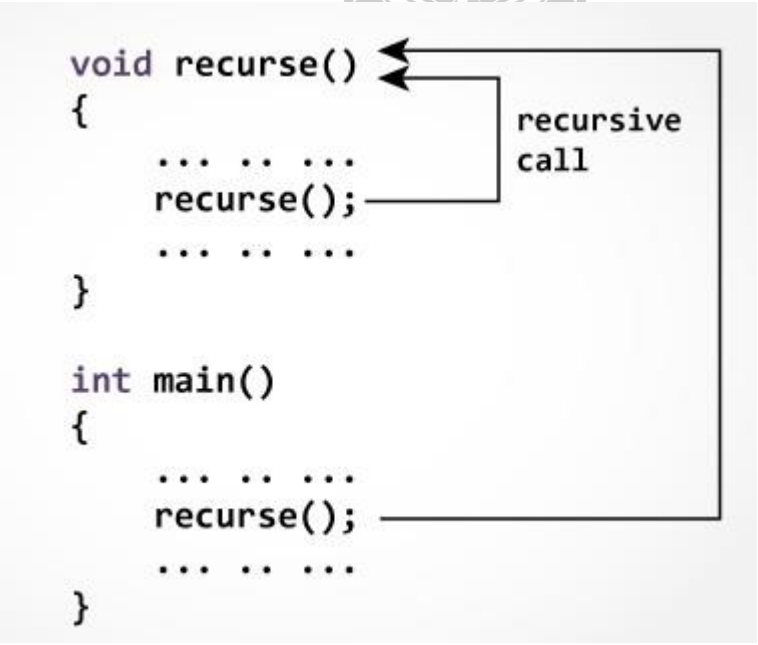
A function that calls itself is known as a recursive function. And, this technique is known as recursion.

Advantage of Recursion

- Function calling related information will be maintained by recursion.
- Stack evaluation will be take place by using recursion.
- In fix prefix, post-fix notation will be evaluated by using recursion.

Disadvantage of Recursion

- It is a very slow process due to stack overlapping.
- Recursive programs can create stack overflow.
- Recursive functions can create as loops.

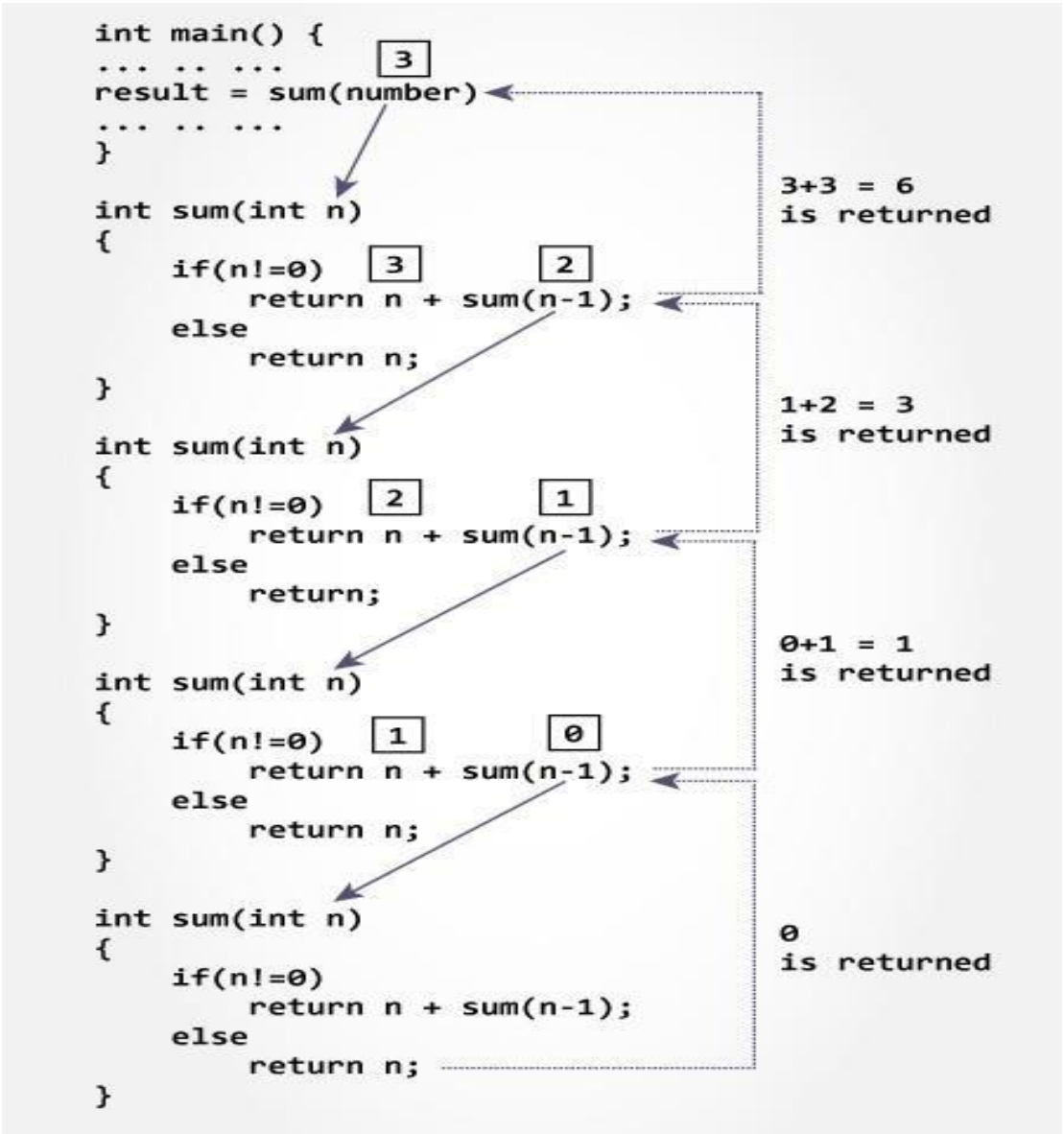


Example: Sum of Natural Numbers Using Recursion

```
#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;
}
```

Output

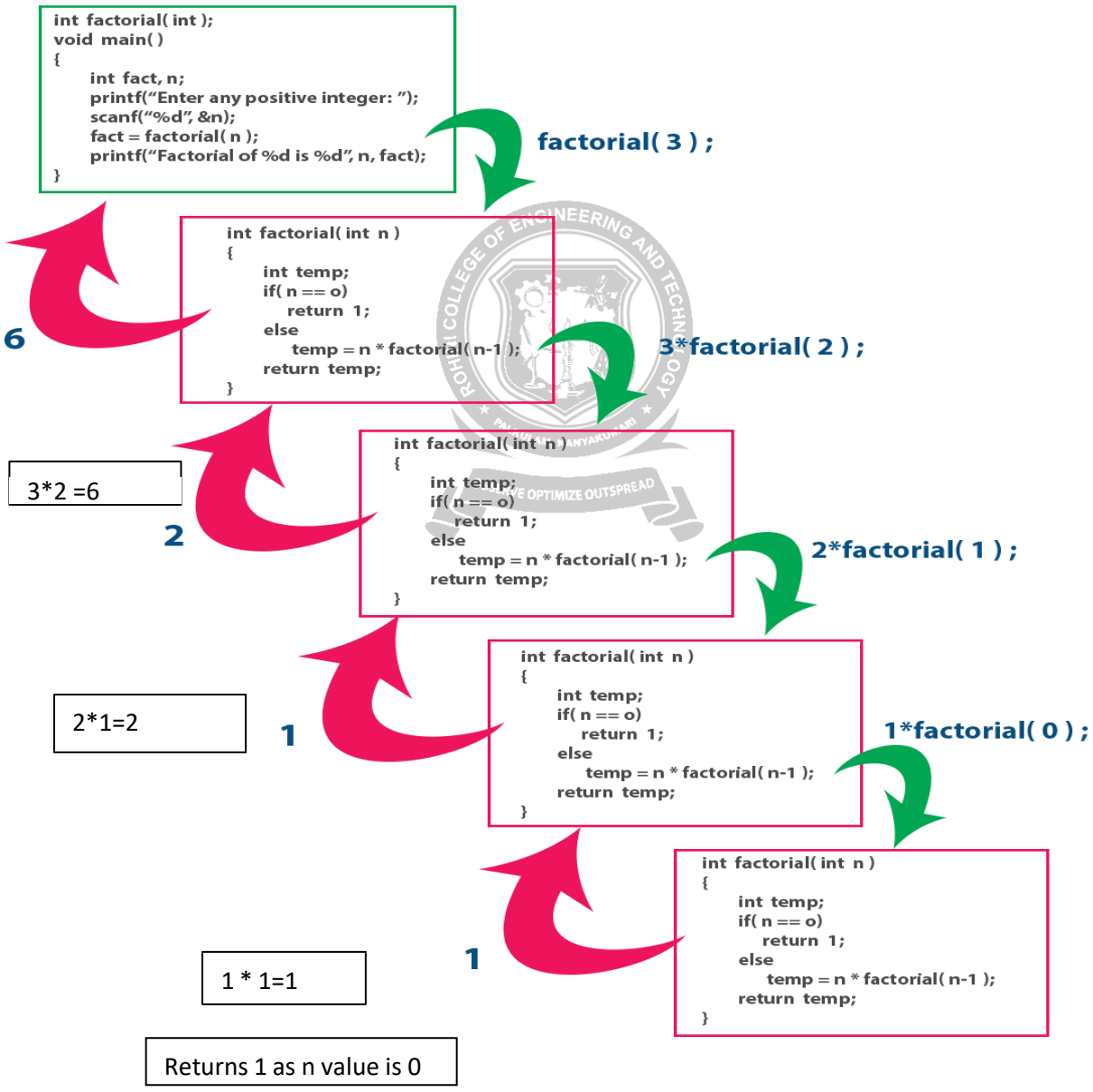
```
Enter a positive integer:3
6
```



FACTORIAL OF A NUMBER USING RECURSIVE FUNCTION

```
#include <stdio.h>
int factorial( int ) ;
void main( )
{
        int  fact, n ;
        printf("Enter any positive integer: ") ;
        scanf("%d", &n) ;
        fact = factorial( n ) ;
        printf("Factorial of %d is %d", n, fact) ;
}
int  factorial( int n )
{
        int temp ;
        if( n == o)
            return 1 ;
        else
            temp = n * factorial( n-1 ) ; // recursive function call
        return  temp ;
}
```

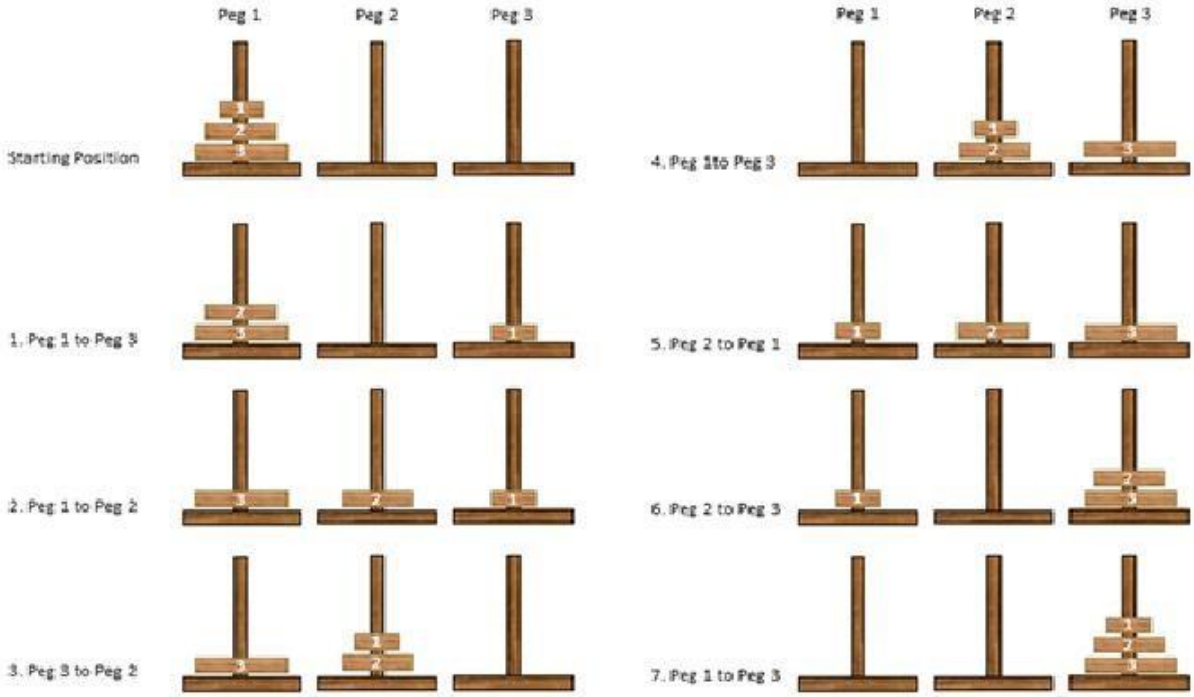
Enter any positive integer: 3
Factorial of 3 is 6



TOWER OF HANOI **OF A NUMBER USING RECURSIVE FUNCTION**

```
#include <stdio.h>
void hanoi(int n, char from, char to, char temp)
{
   if (n == 1)
   {
      printf("\n Move Disk 1 from  Peg %c to %c", from, to);
      return;
   }
   hanoi(n-1, from, temp, to);
   printf("\n Move disk %d from rod %c to rod %c", n, fr, tr);
   hanoi(n-1, temp, to, from);
}
 int main()
{
   Printf(" Towers of Honoi");
   int n;
   printf("\n Enter number of Disks");
   scanf(" %d ",&n); // n implies the number of discs
   hanoifun(n, 'A', 'C', 'B');  // A, B and C are the name of Peg
   return 0;
}
```

To Transfer from disks from peg A to C taking help of B



Output

- Move Disk 1 from Peg 1 to Peg 3.
- Move Disk 2 from Peg 1 to Peg 2.
- Move Disk 1 from Peg 3 to Peg 2.
- Move Disk 3 from Peg 1 to Peg 3.
- Move Disk 1 from Peg 2 to Peg 1.
- Move Disk 2 from Peg 2 to Peg 3.
- Move Disk 1 from Peg 1 to Peg 3.

Example: GCD of Two Numbers using Recursion

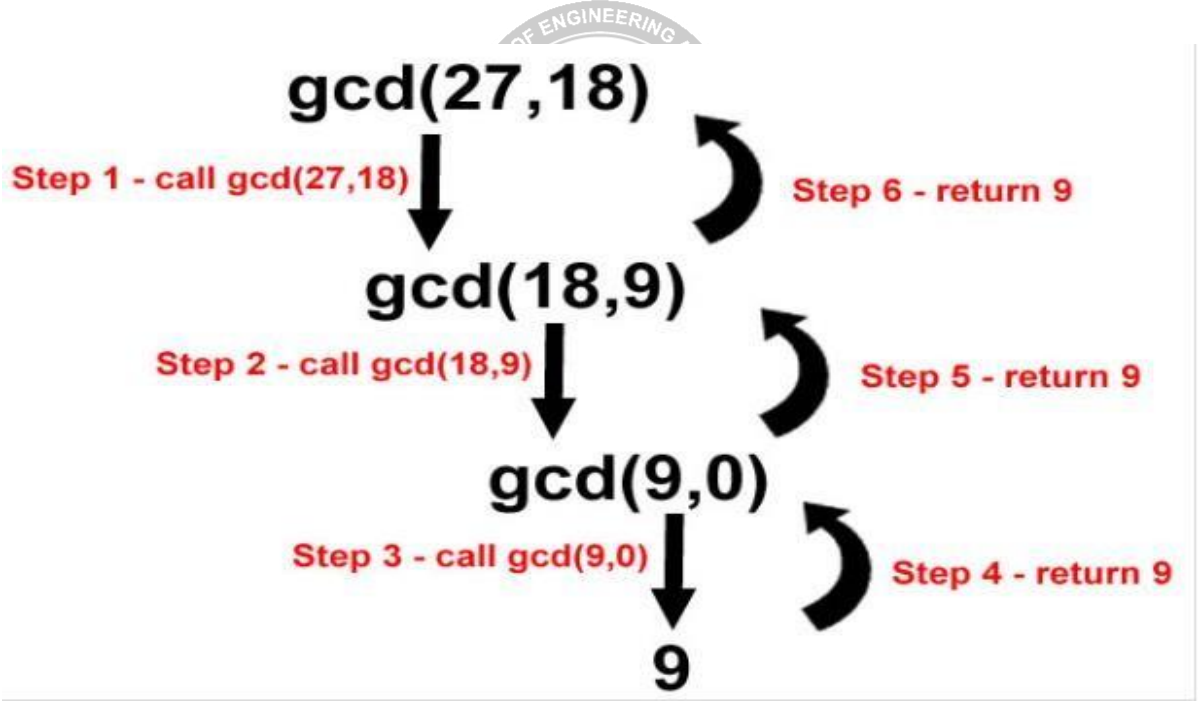
```
#include <stdio.h>
int hcf(int n1, int n2);
int main()
{
   int n1, n2;
   printf("Enter two positive integers: ");
   scanf("%d %d", &n1, &n2);

   printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1,n2));
   return 0;
}

int hcf(int n1, int n2)
{
    if (n2 != 0)
       return hcf(n2, n1%n2);
    else
       return n1;
}
```

Output

```
Enter two positive integers: 27  18
G.C.D of 27 and 8 is 9.
```



Example: Fibonacci Series using Recursion

```
#include
<stdio.h>
int
fibonacci(i
nt i)
{
   if(i == 0)
   {
      return 0;
   }
   if(i == 1)
   {
      return 1;
   }
   return fibonacci(i-1) + fibonacci(i-2);
}

int  main()
{
   int i,f;
   for (i = 0; i < 10; i++)
   {
     f=
     fibonacci(
     i);
     printf("%d
     \n ",f);
   }
   return 0;
}
```

Output

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