

Recursion

Recursion in computer science is a method where the solution to a problem depends on solutions to smaller instances of the same problem. The approach can be applied to many types of problems, and is one of the central ideas of computer science.

A recursive function in C is a function that calls itself. Recursion can be direct recursion or indirect recursion

```
Fn_A()
{
    Fn_A();
    ....
}
```

Direct Recursion

```
Fn_A()
{
    Fn_B();
    ....
}

Fn_B()
{
    Fn_A();
    ....
}
```

Indirect Recursion

Our First Real Example: Factorial

What kinds of problems are well solved with recursion? In general, problems that are defined in terms of themselves are good candidates for recursive techniques. The standard example used by many computer science textbooks is the factorial function.

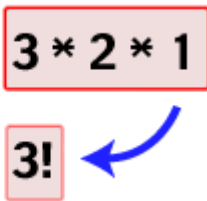
The factorial function, often denoted as $n!$, describes the operation of multiplying a number by all the positive integers smaller than it. For example, $5! = 5*4*3*2*1$. And $9! = 9*8*7*6*5*4*3*2*1$.

Take a good close look at the above, and you may notice something interesting. $5!$ can be written much more concisely as $5! = 5*4!$.

$$5! = 5 * 4 * 3 * 2 * 1$$
$$= 5 * 4!$$

$$5! = 5*4*3*2*1 = 5*4!$$

And $4!$ is actually $4*3!$.

$$4! = 4 * 3 * 2 * 1$$
$$= 4 * 3!$$


$$4! = 4*3*2*1 = 4*3!$$

We now see why factorial is often the introductory example for recursion: the factorial function is recursive, it is defined in terms of itself. Taking the factorial of n , $n! = n*(n - 1)!$ where $n > 0$.

Coding Factorial

Let's try writing our factorial function `int factorial(int n)`. We want to code in then $n! = n*(n - 1)!$ functionality. Easy enough:

```
int factorial(int n)
{
    If (n<=1)
        Return 1;
    return n * factorial(n-1);
}
```

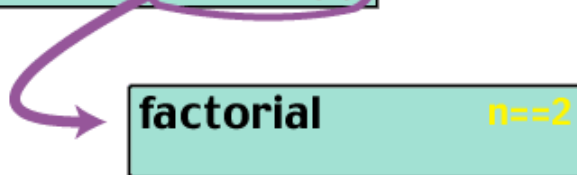
Wasn't that easy? Lets test it to make sure it works. We call factorial on a value of 3, `factorial(3)`:

factorial $n==3$

$$3! = 3 * 2!$$

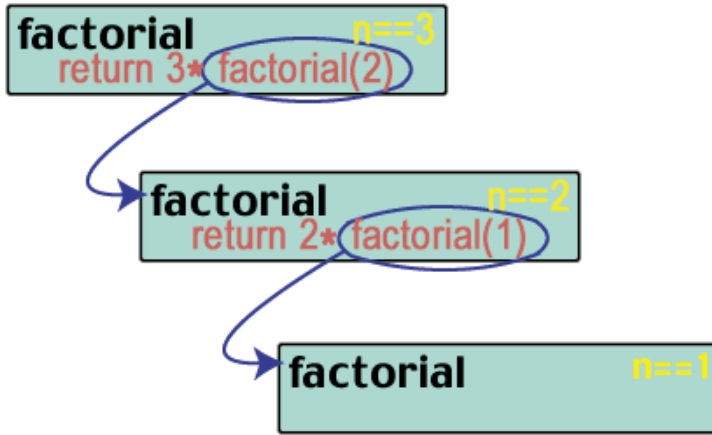
`factorial(3)` returns $3 * \text{factorial}(2)$. But what is `factorial(2)`?

factorial $n==3$
return $3 * \text{factorial}(2)$



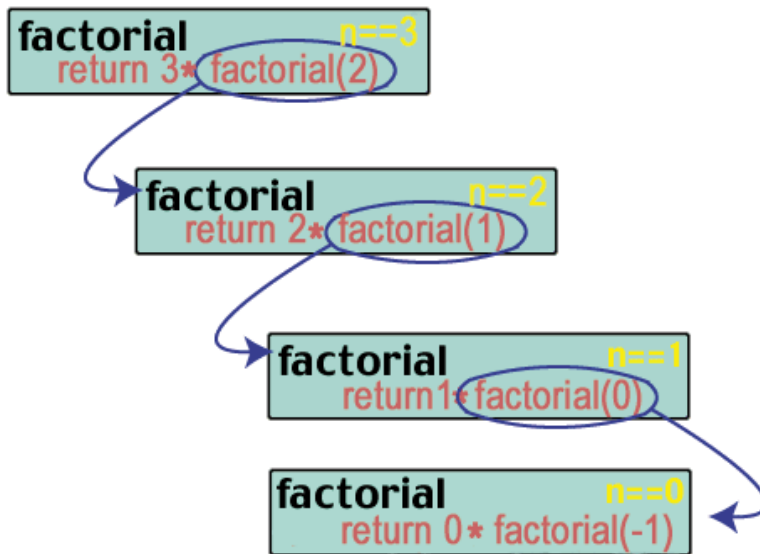
$$2! = 2 * 1!$$

`factorial(2)` returns $2 * \text{factorial}(1)$. And what is `factorial(1)`?



$$1! = 1 * 0!$$

factorial(1) returns 1 * factorial(0). But what is factorial(0)?



$$\begin{aligned}
 \text{factorial}(3) &= 3 * \text{factorial}(2) \\
 &= 3 * 2 * \text{factorial}(1) \\
 &= 3 * 2 * 1 * \text{factorial}(0)
 \end{aligned}$$

As $n \leq 1$ will return 1, we have fact (1) or fact(0) as 1. Here is a another example for a recursion, to find the Fibonacci series.

```

Int function fib(int n)
{
  if (n <= 1)
    return n;
  else
    return fib(n-1) + fib(n-2);
}
  
```