

Section 1 - recursion

Let it be said right from the start:

Recursion in assembly language is not the ball of joy that it is in higher level languages.

Fibonacci Sequence

The Fibonacci Sequence is one primary example of recursion. Consider this example:

```
unsigned int FibonacciInC(unsigned int nthNumber) {  
    if (nthNumber <= 1) {  
        return nthNumber;  
    }  
    return FibonacciInC(nthNumber - 1) + FibonacciInC(nthNumber - 2);  
}
```

The terminal condition is `nthNumber` less than or equal to 1. If the parameter is less than or equal to 1, the very same value is returned.

If the parameter is greater than 1, the result is the sum of the results of the Fibonacci value of one less than the parameter plus the results of two less than the parameter.

Recursion Leverages the Stack

Notice that recursion is accomplished by simply calling a function inside itself. All of the work of implementing recursion is hidden behind the scenes by the compiler.

Specifically, `nthNumber` and the results of `FibonacciInC(nthNumber - 1)` is pushed onto (and popped off of) the stack. The value of `FibonacciInC(nthNumber - 2)` doesn't go on the stack because its results are used right away.