

Formula

This program is to expand the polynomial $(1+x)^n$, and n is less than or equal to 13. In `formula.c`, it calls `nCr.s` with two elements, the input integer and the degree of the polynomial which is from 1 to input integer, and the `nCr.s` return "+ coefficient x^{degree} ". In `nCr.s`, there's 2 functions, factorial and main method. Factorial function simply takes a number and return the factorial of that number by keep multiply until the parameter number is less than 1. (think in recursion) In `nCr` function, it takes 2 parameters, n and r , and returns the total numbers of ways to choose r elements from n elements. The function is $n!/r!(n-r)!$. Because it is easily reach the overflow, so I try to cancel the common denominator, $n \cdot n-1 \cdot \dots \cdot r/r!$ or $n \cdot n-1 \cdot \dots \cdot n-r/(n-r)!$. In this case, the function reaches the overflow at $n \geq 13$. So I wrote an if statement to make sure n will be less than 13. If n bigger or equal to 13 the main function will output error message. Also the another error message will print out if input is less than 0.

Notice that every time the time require will be different based on the memory usage and the CPU(I guess...)

Big O for factorial function is $O(n)$ since the function just simply use the recursion. Big O for `nCr` function is $O(n)$, it calls the factorial function and find a constant from the return value of factorial function. Big O for `formula.c` is $O(n^2)$, it calls the `nCr` function n times, and the big O for `nCr` is $O(n)$ so $n \cdot n = n^2$.

Mystery

The mystery program can compute the Fibonacci sequence

Ex:

$$f(1)=C(0,0)=1$$

$$f(2)=C(1,0)=1$$

$$f(3)=C(2,0)+C(1,1)=1+1=2$$

$$f(4)=C(3,0)+C(2,1)=1+2=3$$

$$f(5)=C(4,0)+C(3,1)+C(2,2)=1+3+1=5$$

$$f(6)=C(5,0)+C(4,1)+C(3,2)=1+4+3=8$$

$$f(7)=C(6,0)+C(5,1)+C(4,2)+C(3,3)=1+5+6+1=13.$$

My `mystery.c` based on the given assembly language. My `mystery.c` contain three functions. Add function, dothething function and main function. In the add function, it plus two integers. In the dothething function, it use recursion to make it add from 1. My main integer is just pass the number in it. Also I add a bit more condition, if input is 0, then it will print 0, if input is smaller than 0, it is invalid input.

Thanks for reading,

Have a great day.