Design

- takes a positive integer and prints out the expanded form
- utilizes two programs formula written in C and nCr written in assembly

Implementation

- Factorial
    - simple function that runs in O(n) time cause its called n times
- nCr
    - does the basic arithmetic on the given inputs calling factorial to create the various components of the function
    - runs in O(n) time due to three factorial calls and since factorial is O(n), 3*O(n) is still O(n)
- Formula
    - Formula runs n times calling nCr n time and since nCr is O(n), n*O(n) is O(n^2)

Space

- the most space this whole program takes is create the variables and initializing them n time, so space would n since there is a multiple of n variables being made locally in the program

Problems

- since the program had to be 64bit, i found that any number n greater than 65 caused an overflow error, hence i remedied it by inputting a conditional guaranteeing no input greater than 65 can be allowed