For the nCr.s file I began by thinking of my code in terms of C. I determined what was needed in order to ensure that the program would run successfully. I began by creating my factorial method which was a basic for loop calculating the factorial. My nCr method was a bit more complex. I continuously concatenated to a string with the integer. I first had to convert the user input to an integer using the atoi function, then I proceeded by doing the combination with the user input and exponent and concatenated my string and kept doing this until the exponent was equivalent to the user input. The Big O run time was O(n) because everything was being done in linear time. It depended on the user input and would do that many calculations repeatedly. The code was optomized by condensing the list of instructions needed to do the computations. The performance of the program was good in the sense that if a large factorial wasn't provided then the program would run fairly quickly. Some challenges I personally faced was being able to concatenate the string because I had to account for the memory allocation. However, after researching I used a sprtintf() function which handled the memory allocation as well as the concatenation. Overall the program took in one parameter, which was then used to determine the expanded for of an equation (1+x)^n. The user would provide the exponent, and the program would return the expanded form. The professor informed the class to not worry about extraneous conditions.