

Hiren Patel

Readme.pdf

Formula

The implementation of the program is simple. Formula runs a while loop to get the coefficients by calling nCr function which in turn calls the Factorial function. The Factorial function only takes one parameter and since it uses an integer, can only go up to 12! Before giving you a wrong value. This is explicitly stated within the -h help command that we were told to write. The instructions said to return a zero value for any overflow that might happen, which I did. This however, results in a floating point error since dividing by zero isn't allowed. The factorial function coupled with the nCr function is not very efficient. It simply calculates the factorial for each value and then divides the resultant by the formula given. A much more efficient use would be to send two parameters, the first being the smaller factorial we will eventually be dividing by and the second being the input/ numerator factorial(n). The smaller factorial we will be dividing by can be found by first comparing which is bigger, r or (n-r). Calling the bigger value x (which has to be smaller than n) and then understanding that $n!/x! = (x+1)*(x+2)*(x+3)*\dots*n$, we could much more efficiently calculate the quotient of two factorials doing less work than we are currently doing with one factorial. This would GREATLY decrease the time to run the program. I would have implemented this, however, we are given a Factorial function with only one parameter.