Aman Singh

Formula Readme

Design/Challenges

Challenges

The biggest obstacle I faced was writing assembly for Factorial and nCr. Thinking of how they would be implemented in C helped.

Design

As mentioned before Factorial and nCr were simple to write when thinking in terms of C representation. Overflow of above 12 returned a 0. Formula converts the input from the user into a string and executes depending on the number:

1) Input is bigger than 12 it will result in an overflow error
2) Input is negative it'll tell them to use the help flag
3) Input is –h display help message
4) Input are letters, characters, etc. returns 1. Unless first few characters are numbers less than 12 followed by letters or characters, ie 12dqre14f- will give the binomial expansion of 12.

If none of these conditions apply then it will calculate the binomial expansion printing strings for '+', '*x', and '^', using the nCr function to calculate the coefficients, and a loop for the exponent. Run time is found using gettimeofday().

Analysis

Factorial and nCr are written for a 64 bit processor as the professor said we could, but 32 bit can be provided if necessary. Because the input is a 32-bit integer, the program can only store up to 2^31-1. Therefore the maximum value one can input is 12, or else an overflow error will occur. Using other data types will increase this maximum value. My formula program runs in O(n).