Rob Henderson
Rutgers University
TA: Binh Pham
cs211 Comp Arc

int PrintBinomialTheorum(int n) is the function that prints out the binomial theorum.
PrintBinomialTheorum calls nCr (located in the .s file) inside of a for loop.

```
for (r = 0; r <= n; r++) {
nChooseR = nCr(n, r);
........
}
```

nCr calls Factorial 3 times with n,n-r, and r since nCr= ((n!)/(n-r)!)/r!.

Design challenges included getting nCr to be able to call Factorial. Then we needed to be able to identify error conditions and how to handle them. In particular, the overflow check in Factorial was important. The check for overflow had to be done after the multiplication is performed. If overflow is detected, the function needs to return 0 to signify that (since no valid input will ever result in 0). From there, nCr has to know what to do with a bad value generated from Factorial. Basically, if we know that n! generates an overflow we know that there is a problem and we need to just return 0 in the nCr function. From there, PrintBinomialTheorum is printing out the nCr values 1 at a time in the loop and if nCr is 0, it reads out an error to stderr (since no valid input will result in 0).

Error checking is also very important. I decided to use strtol instead of atoi becuase of its ability to detect bad input. For bad input (non integers), strtol is setting a string to NULL to signify that an error occurred. If an error occurred or it parsed the input to a negative number, I didnt even bother to go into the PrintBinomialTheorum.

Big-O   Analysis

This program is O(n) in times of space and performance. Assuming that the input n is valid, for (1+x)^n, as n increases linearly. PrintBinomialTheorum gets executed 1 more time every time n is increased by 1. As a result of PrintBinomialTheorum getting called 1 more time, nCr gets called 1 more time, and Factorial gets called 1 more time.