Formula.c takes in a user nonnegative integer input (up to 13) and prints out the “long” form of (1+x)^n where n is the user input. There are three parts to formula.c, factorial(n) which uses tail recursion to solve for n!, nCr(n,r) solves for nCr by calling factorial(n), factorial(r) and factorial(n-r). My main calls nCr from r = 1 till r = n. Tail recursion cut down my run time from 50-40 microseconds down to 30-20 microseconds. In terms of space analysis, my program is small, main calls nCr which in turns calls factorial. Factorial would take up a lot of space if tail recursion had not been used. BigO is n^2 since the call is recursive.