Read Me

For programs 1-4

For detailed explanations refer to the code itself comments appear for blocks of new code

All of these programs were done using windows 7 and devC++ on a 64 bit processor just to make sure it would run on a x86 processor long long integers were removed.
The make file given is for windows although program 4 comes with a project file in order to run.
The rest of the programs 1-3 are stand alone and do not require make files.

Program 1: publicPrivateGen.cpp
only uses the c++ libraries
The idea is to take in user input on the comand line either (prime1, prime2 (-o filename)*)
()* not required
or (-c (-o filename))
once the prime numbers are given or generated they are used to create the rsa keys
(e, n) and (d, n)
the math behind the calculations is in the program

Progam 2: asciiToDecimal.cpp
converts an ascii file by traversing each character in every line and outputting the decimal values to a new file which is prenamed in a line by line fashion.

Program 3: decimalToAscii.cpp
does the inverse of program 2 and takes the line and converts the value back into an ascii character then places them onto one line line breaks were created in program 2 so these correspond to the line breaks needed for the ascii transformation

Program 4: encryptDecrypt.cpp
This program is part of a project that was done on windows so the make file is oriented towards windows I was unable to get the make file working for unix machines.
The reason a make file is needed is that this program has a dependency upon a class called BigInteger. This class is not my own creation it was created by Matt McCutchen and his read me is included with program 4 in the folder. All the files needed to run program 4 are in the folder called program 4. Along with the make file skeleton for unix as stated it was not completed only the windows make file is. This program takes a Decimal file and encrypts/decrypts it. The program knows neither what encryption or decryption is only the math behind the key. Encryption and Decryption use the same mathematical formula which was given in the BigInteger class. To see this function refer to the BigInt readme. Once the number is run through the mathematical process this number saved in a new file. The file follows the form (original name) + NEW + .txt a new name can be given on the command line with the -o command. The contents of the original file are not modified just in case the user input the wrong key with a proper file. The resulting file would be garbage and the user would have to rerun the program with the proper key on the original. Once this is done program 3 can be used to return the file back to its original ascii values assuming decryption occurred else the values will be to great to conver to ascii.