# Assignment 5 - Implement the Public Key Encryption Scheme: RSA

## 1 Tasks to be Performed

- Implement the RSA key generation function that takes a positive integer $k$ as an input and outputs (1) a file with $(p, q)$, (2) a file with the the public key $(N, e)$, and (3) a file with the private key $(N, d)$, such that $N$ is a $k$-bit integer. These three files should be saved with numbers in decimal format, 1 per line in three text files (the two keyfiles to be imported below).
- Implement the RSA encryption and decryption functions

When the main program is executed, here is the expected output:

1. Enter the bit size:
2. Enter the name of the public key output file to save:
3. Enter the name of the private key output file to save:
4. Enter the name of the p q output file to save:
5. Enter the name of the public key file to import: (Note: This could be either the file you generated above, or a file we provide, and must be UTF-8 Unix delimeted).
6. Enter the name of the file that contains $x$ to be encrypted using $(N, e)$: (Note: this file will be UTF-8 Unix delimeted text, which you will need to convert to the appropriate numeric type; an example is provided)
7. Enter the output file name to store $E(x)$, which is $c$:
8. Enter the name of the private key file to import: (Note: This could be either the file you generated above, or a file we provide, and must be UTF-8 Unix delimeted).
9. Enter the name of the file that contains $c$ to be decrypted using $d$:
10. Enter the output file name to store $D(c)$:

## 2 Programming Language and Library Requirements

This project needs to be implemented in C++ and uses the GMP library (The GNU Multiple Precision Arithmetic Library, http://gmplib.org/) to manipulate big numbers.

## 3 Deliverables

- README: describe the purpose of your files and provide instructions on how to compile and execute your program.
- Well-documented source code.