Introduction to Computer Science
HW #3
Due: 2019/04/11

Homework Rules:

Hand-written homework can be handed in **before lecture starts**. Otherwise, you may contact the TA in advance and then bring the hardcopy to the TA in MD-631 (please send e-mail in advance).

As for the programming part, you need to upload it to CEIBA before the deadline (2019/04/11 3am). You can choose to program in C++ or in Python. The file you upload must be a **.zip** file that contains the following files:

README.txt

HW03_b07901XXX (a folder that contains all .cpp & .h (or .py) as required),

If you program in C/C++:

1. Do not submit executable files (.exe) or objective files (.o, .obj). Files with names in wrong format will not be graded. You must **remove any system calls**, such as system ("pause"), in your code if any.
2. In README.txt, you need to describe which compiler you used in this homework and how to compile or execute it (if it is in a "project" form).
3. In your .cpp files, we suggest you write comments as detailed as you can. If your code does not work properly, code with comments earns you more partial credits.

If you program in Python:

1. Do not submit .pyc files. Files with names in wrong format will not be graded.
2. In README.txt, you need to describe which Python version you used in this homework (e.g. Python 3.5) and how to execute it.
3. In your .py files, we suggest you write comments as detailed as you can. If your code does not work properly, code with comments earns you more partial credits.

Introduction to Computer Science
HW #3
Due: 2019/04/11

Chapter 4 Review Problems (6 pts each)

22, 25, 29, 40

Chapter 5 Review Problems (8 pts each)

16, 39, 50, 53

Programming Problem (44%)

First, VERY IMPORTANT: check whether sizeof(unsigned long long int) or sizeof(unsigned long int) is 8. If not, use another computer.

Write three pieces of code:

(a) cipher.cpp/cipher.py reads the file "plain.txt" containing one string (length < 10000) and "public_key.txt" containing *N* and *e*. cipher.cpp/cipher.py should then output "secret.txt" as integers encrypted by RSA. The encoding concatenates 2 chars into one integer. For example, "AB" would be encoded as ($65*2^8+66$)=16,706. If only one char remains, put it to leftmost. For example, "A" would be encoded as $65*2^8$=16,640.

(b) decipher.cpp/decipher.py reads the file "secret.txt" and "private_key.txt" containing *N* and *d*. decipher.cpp/decipher.py should then output "message.txt" with content same as "plain.txt".

(c) find.cpp/ find.py reads e and phi from file "cryptan.txt" and output "cryptan_result.txt" with d where $de \equiv 1 \pmod{phi}$ using Euclidean algorithm.

Note: Be careful about overflow, signed/unsigned, and eof() problem. "Npqphied.txt" contains two more (*N*,*e*,*d*,*phi*) sets for you to test.

Note2: Any libraries containing the function with RSA cipher and decipher or the function with Euclidian algorithm are prohibited.

Introduction to Computer Science
HW #3
Due: 2019/04/11

Bonus (5%)

Write bonus.cpp which is capable to do matrix multiplication. It should be able to read "matrix_in.txt" with matrix size *n* and two *n* by *n* matrices, denoted by A and B and output $A \times B$ to "matrix_out.txt". All the numbers in matrix are floating point numbers in the range $(-1,1)$. Input format would be like:

```
2
0.1 0
0 0.1

0.1 0
0 0.1
```

While the output should be:

```
0.01 0
0 0.01
```

Your code needs to handle *n*=1000 or even more in reasonable time (n^3 algorithm is not acceptable). You are not allowed to use any library of function that can do matrix multiplication.

* Strassen algorithm :
https://ccjou.wordpress.com/2013/06/04/%E5%88%86%E6%B2%BB%E7%9F%A9%E9%99%A3%E4%B9%98%E6%B3%95%E2%94%80%E2%94%80strassen-%E6%BC%94%E7%AE%97%E6%B3%95/