1) Determine the following values:

a) $\Phi(7502752)$
$= \Phi(2^5)\Phi(3^5)\Phi(7)\Phi(17)$
= (2^5 - 2^4)(3^5 - 3^4)(7 - 1)(17 - 1)
= 1,990,656

b) $\Phi(11165)$
$= \Phi(5)\Phi(7)\Phi(11)\Phi(29)$
= (5 - 1)(7 - 1)(11 - 1)(29 - 1)
= 6,720

c) $\Phi(123456789)$
$= \Phi(3^2)\Phi(3607)\Phi(3803)$
= (9 - 3)(3607 - 1)(3803 - 1)
= 82,260,072

d) $\Phi(69500000)$
$= \Phi(2^5)\Phi(5^6)\Phi(139)$
= 2^5 5^6 139
= (32 - 16)(15625 - 3125)(139 - 1)
= 27,600,000

e) $\Phi(10967535067)$
$= \Phi(104723)\Phi(104729)$
= (104723 - 1)(104729 - 1)
= 10,967,325,616

2) Without the aid of any computing device, show how one can use Euler's Theorem to determine the remainder when 6^{15604} is divided by 1925.

Find: 6^{15604} % 1925
$\Phi(1925)$
$= \Phi(5^2)\Phi(7)\Phi(11)$
= 1200
$6^{1200} = 1 \bmod 1925$
$6^{15604} = (6^{1200})^{13} * 6^4 \bmod 1925$
$= 1 * 6^4 \bmod 1925$
$= 6^4 \bmod 1925$
$= 1296 \bmod 1925$

3) Random bits algorithm
See `h3.3_1.py` and `h3.3_2.py`
Usage: Run `h3.3_1.py` to generate our 20k random bits, and `h3.3_2.py` to run the tests on our output file from the first script.

Part 1
a) Pick a random prime number, p, with p > 10^6.
b) Randomly pick integers until you identify a generator for this prime number, g.
c) Use a time function on your system to return a value (in C this is seconds past Epoch time (midnight 1/1/1970)) and then take this value mod 100. Call this s.
d) Calculate x = g^s mod p
e) Generate 20,000 'random' bits by:
(g^x mod p) mod 2, (g^(x+1) mod p) mod 2, …, (g^(x+19999) mod p) mod 2.
Then use these 20000 bits to see if they pass the random bits test:
Part 2
a) The number of 0s and 1s in the range [9654, 10346].
b) If we break the 20000 bits into 5000 blocks of 4 bits and interpret these values from 0-15, letting n_i (0 <= i < 16) represent the number of blocks with value i, and let

$$X = \frac{16}{5000} \sum_{i=0}^{15} n_i^2 - 5000$$

The test is passed iff 1.03 < X < 57.4

c) Split the 20000 strings into maximal runs of 0s and 1s. The number of runs of each length should be in the following ranges:

Length of Run	**Required Interval**
1	2267 - 2733
2	1079 – 1421
3	502 – 748
4	223 – 402
5	90 – 223
6 or more	90 – 223

4)
See `Q4.java`

x = 128567, y = 337031

$M^1 = M^{1106427x + 422068y}$
$= M^{1106427x} * M^{422068y}$
$= 1106427^{128567} * 422068^{337031}$
$= 77453$

5)
p = 12346123784612378461291
g = 97997897867
b (bob's secret number) = 12341234123432431355555
message from alice to bob: 9583793135283340325422
Bob sends to Alice
g^b mod p
$= 97997897867^{12341234123432431355555} \bmod 12346123784612378461291$
$=10301296542184976843387$
Secret Key is
$9583793135283340325422^{12341234123432431355555} \bmod 12346123784612378461291$
$= 4147209013993051259287$

6)
See `Q6.java`
Try to break this message. Numbers of interest are: 926002123823977 and 3156478917144031.
299895574003382
2781819096562037
2870170500708305
600200268479477
Looking this over, this is RSA. n = 3156478917144031, e = 926002123823977
Decryption exponent: 2285132565328881
Decrypted texts:
36159439518164
92961002036097
36603649729660
33551417261566