Problem G

RSA Cryptosystem

Time limit: 3 seconds

Memory Limit: 1048576 bytes

The RSA cryptosystem, named after its inventors Ronald Rivest, Adi Shamir and Leonard Adleman, at MIT in 1977, is the most widely used public-key cryptosystem. It may be used to provide both secrecy and digital signatures and its security is based on the intractability of the integer factorization. The RSA cryptosystem has been patented in the U.S. and Canada. Several standards organizations have written standards that address the use of the RSA cryptosystem for encryption, digital signatures, and key establishment. An example of implementation of RSA is Secure Socket Layer (SSL) applied to HyperText Transfer Protocol, which is known as "HTTP Secure" or HTTPS [1].

Suppose Bob wants to send a message m to Alice. Firstly, Alice has to create an RSA public key and a corresponding private key, by performing Algorithm 1 [2]. Secondly, Bob encrypts the message m for Alice, which she decrypts, by performing Algorithm 2 [2].

Algorithm 1: Key generation for RSA public-key encryption

1) Generate two large random and distinct primes p and q.
2) Compute $n = pq$ and $\varphi = (p - 1)(q - 1)$.
3) Select a random integer e, $1 < e < \varphi$, such that $\gcd(e, \varphi) = 1$.
4) Compute the unique integer d, $1 < d < \varphi$, such that $ed \equiv 1 \pmod{\varphi}$ using extended Euclidean algorithm [3].
5) Alice's public key is (n, e); Alice's private key is d.

Algorithm 2: RSA public-key encryption

Encryption. Bob should do the following:

1) Obtain Alice's authentic public key (n, e).
2) Represent the message as an integer m in the interval $[0, n - 1]$.
3) Compute $c = m^e \bmod n$.
4) Send the cipher text c to Alice.

Decryption. To recover plaintext m from c, Alice should do the following:

1) Use the private key d to recover $m = c^d \bmod n$.

However, a plaintext message m, $0 \le m \le n - 1$, in the RSA public-key encryption scheme is said to be unconcealed if it encrypts to itself; that is, $m^e \equiv m \pmod{n}$. There are always some messages which are unconcealed. An issue when choosing e is that there should not be too many unconcealed messages. An example is when $p = 37$ and $q = 19$ then $n = 37*19 = 703$ and $\varphi = 36*18 = 648$. If we choose $e = 181$, although $\gcd(181, 648) = 1$, you may check that it turns out that all possible messages m ($0 \le m \le 702$) are unconcealed when computing values $m^e \bmod 703$. For any valid choice of e there exist some unconcealed messages. It is important that the number of unconcealed messages is at a minimum. For given values of p and q, define S as the sum of all values of e, $1 < e < \varphi$ and $\gcd(e, \varphi) = 1$, so that the number of unconcealed messages for this value of e is at a minimum. For example, when $p = 41$ and $q = 53$ then $S = 274952$. Your task is to compute S for the given values of p and q.

References:

[1] *https://tiptopsecurity.com/how-does-https-work-rsa-encryption-explained*, retrieved 15/8/2018.

[2] Menezes et al., *Handbook of Applied Cryptography*, CRC Press, 1996.

[3] Cormen et al., *Introduction to Algorithms, 3rd Edition*, The MIT Press, 2009.

Input:
Values of the primes p ($1 < p < 4096$) and q ($1 < q < 4096$)

Output:
S

Sample Input 1: 41 53 **Sample Input 2:** 1117 1291	**Sample Output 1:** 274952 **Sample Output 2:** 46214815518