RSA key generation

① Select at random two large prime numbers p, q.

② calculate n; $n = p * q$; n becomes the modulus.

③ Calculate $\phi(n)$ for n.
For the n; $\phi(n) = (p-1) \times (q-1)$; This is called euler's totient function

$\phi(n)$ calculates the number of positive integers less than or equal to n that are coprime with n.

④ choose an integer e (encryption key) such that $1 < e < \phi(n)$ and e is coprime to $\phi(n)$.

⑤ compute d (decryption key) as the modular inverse of e with respect to $\phi(n)$.

⑥ use

1. (e, n) as public key
2. (d, n) as private key

Def: A is coprime/relatively prime to B, If A & B have no common factors other than 1.

7 & 1 are coprime $GCD(7,1)=1$

7 & 2 are coprime $GCD(7,2)=1$

8 & 4 are not coprime $GCD(8,4)=4$

9 & 7 are coprime $GCD(9,7)=1$

If p, q are prime
p & q are coprime

If p, q are even
p, q are not coprime.

Def: Modular inverse of number a with respect to n is the number b such that the product of a and b is congruent to 1 modulo n

$$a \cdot b \equiv 1 \pmod{n}$$

Basically mean

$$[a \cdot b \bmod n] = [1 \bmod n]$$