- Where g is a primitive root of p.
- Let p be a prime. Then g is a *primitive root* for p if the powers of g, 1, g, g^2, g^3, ... include all of the residue classes mod p (except 0)
- **Examples:**

If p=7, then 3 is a primitive root for p because the powers of 3 are

1, 3, 2, 6, 4, 5---that is, every number mod 7 occurs except 0.

But 2 isn't a primitive root because the powers of 2 are

1, 2, 4, 1, 2, 4, 1, 2, 4...missing several values.

- **Example:**

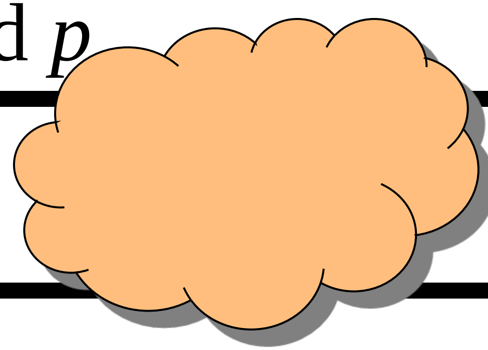
If p=13, then 2 is a primitive root because the powers of 2 are 1, 2, 4, 8, 3, 6, 12, 11, 9, 5, 10, 7---which is all of the classes mod 13 except 0.

There are other primitive roots for 13 (?).

g

p

$$A = g^x \bmod p$$



$$B = g^y \bmod p$$

A

B

$$k = B^x \bmod p$$

$$k' = A^y \bmod p$$

$$k' = k = g^{xy} \bmod p$$

- Example : -
- Alice and Bob agree on **p = 23** and **g = 5**.(show that 5 is primitive root of 23)
- Alice chooses **a= 6** and sends **5 6 mod 23 = 8**.
- Bob chooses **b = 15** and sends **515 mod 23 = 19**.
- Alice computes **19 6 mod 23 = 2**.
- Bob computes **815 mod 23 = 2**. Then **2** is the shared secret.
- Clearly, much larger values of **a, b,** and **p** are required.

Rivest, Shamir and Adleman (RSA)

- RSA stands for Rivest, Shamir, and Adleman, they are the inventors of the RSA cryptosystem. RSA is one of the algorithms used in PKI (Public Key Infrastructure), asymmetric key encryption scheme. RSA is a block chiper, it encrypt message in blocks (block by block). The common size for the key length now is 1024 bits for P and Q, therefore N is 2048 bits, if the implementation (the library) of RSA is fast enough, we can double the key size.
- **Key Generation Algorithm**
- Generate two large random primes, *p* and *q*, of approximately equal size such that their product n = pq is of the required bit length, e.g. 1024 bits.
- Compute n = pq and (ϕ) phi = (p-1)(q-1).
- Choose an integer *e*, 1 < e < phi, such that gcd(e, phi) = 1.
- Compute the secret exponent *d*, 1 < d < phi, such that ed ≡ 1 (mod phi).
- The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and phi secret.
- n is known as the *modulus*.
- e is known as the *public exponent* or *encryption exponent* or just the *exponent*.
- d is known as the *secret exponent* or *decryption exponent*.`

- In encryption, represents the plaintext message as a positive integer *m and* computes the ciphertext $C = m^e$ ***mod n.***
- In decryption compute $m = c^d$ ***mod n***
- Example : let p=17 & q=11 then
- Compute n = pq =17×11=187.
- Compute ø(n) or (ϕ) phi =(p–1)(q-1)=16×10=160.
- choose e=7 (1 < e < 160) where gcd(7,160)=1.
- d=23 where 1 < d < 160 and ed ≡ 1 (mod 160).(multiplication inverse).
- The public key is (187, 7) and the private key is (187, 23).
- given message M = 88 (88<187)
- encryption: $C = m^e$ ***mod n***: $C = 88^7$ ***mod 187*** = 11 .
- Decryption: $m = c^d$ ***mod n:*** $m = 11^{23}$ ***mod 187=88.***

Ex/ p=3,q=11,e=7,m=2 encrypt and decrypt using RSA Algorithm?

- Choose p = 3 and q = 11
- Compute n = p * q = 3 * 11 = 33
- Compute $\phi(n) = (p - 1) * (q - 1) = 2 * 10 = 20$
- Let e = 7
- Compute d = 3 [(3 * 7) mod 20 = 1]
- Public key is (e, n) => (7, 33)
- Private key is (d, n) => (3, 33)
- The encryption of $m = 2$ is $c = 2^7 \bmod 33 = 29$
- The decryption of $c = 29$ is $m = 29^3 \bmod 33 = 2$