→ Miller-Rabin Primality Test

→ This method is based on Fermat little Theorem

Revision

→ Fermat little theorem

→ If n is a prime number, then for every a,
$1 < a < n-1$,

$$a^{n-1} \equiv 1 \pmod{n}$$

or

$$a^{n-1} \% n = 1$$

Ex → $n = 5$

$1 < a < (5-1=4)$

~~Let~~ Take $a = 2, 3, 4$

Check

$2^4 \% 5$	$3^4 \% 5$	$4^4 \% 5$
$= 16 \% 5$	$= 81 \% 5$	$= 256 \% 5$
$= \underline{\underline{1}}$	$= \underline{\underline{1}}$	$= \underline{1}$

→ And it is also based on the following:-

→ If x, n are positive integers such that $x^2 \equiv 1 \pmod{n}$ but $x \not\equiv \pm 1 \pmod{n}$ then n is composite.

Proof → Given:- $x^2 \% n = 1$

which means $(x^2-1) \% n = 0$

This means n divides (x^2-1)

$[(x-1)(x+1)]$ — (1)

but $x \% n \neq \pm 1$ — (Given)

$(x-1) \% n \neq 0$ and $(x+1) \% n \neq 0$

which means n doesn't divides $(x+1)$ and $(x-1)$ — (II)

eqⁿ (1) and (2) can't be true simultaneously when n is prime, so n is composite.

→ <u>Algorithm</u>

→ returns false is n is composite and true if n is probably prime.

→ k is input parameter that determines accuracy level.

Higher value of k ⟶ more accuracy

bool isPrime (n, k) :

→ Handle base cases for n<3

→ If n is even, return false

→ Find odd number d such that n-1 is written as $(d \times 2^r)$. since n is odd so (n-1) must be even and r > 0 (must)

→ Do following k times [accuracy factor]

if (millerTest (n, d) == false)
{ return false }
return true;

→ In millerTest (n, d)

→ Pick random number from range [2, n-2]

→ Compute x = Pow (a, d) % n

→ If x == 1 or x == n-1, return true.

→ Do foll. while (d != n-1)

→ x = (x * x) % n

→ if (x == 1) return false;

→ if (x == n-1) return true;

Example :- Input $n = 13$, $k = 2$

→ Computing d and r, such that $d \times 2^r = n-1$

$d = 3$, $r = 2$

→ Call miller test k times

→ 1st iteration

→ Taking a from $[2, n-2]$ $\xrightarrow{}$ 11

$a = 4$

→ compute $x = \cancel{pow}\ a^d \% n$

$x = 4^3 \% 13$

$= 12$

→ $12 == (13-1) \Rightarrow x = (n-1)$

so return true

→ 2nd iteration

→ suppose $a = 5 \in [2, n-2)$

→ $x = a^d \% n$

→ $x = 5^3 \% 13 = 8$

Since $x = 8$ is not equal to 1 or 12;

→ while $(d != n-1)$

→ $x = (x * x) \% 13 = (8 \times 8) \% 13 = 12$

$(d = 2 \times d;)$ → $x == 12$ $(n-1)$ return true

Since both iteration is true, we return true.

→ Some imp. facts:-

1) Fermat theorem → $a^{n-1} \% n = 1$ for $1 <= a < n$

2) Base cases make sure n is odd, so $n-1$ must be ~~ven~~ even and every $(n-1)$ can be written as $d * 2^s$ where $s > 0$ and d is odd

3) From above 2 points, for every randomly picked no. in range $[2, n-2]$, the value of $a^{d * 2^n} \% n$ must be 1.

4) from earlier discussed lemma, ~~for n to be prime~~ if $x^2 \% n = 1$ or $(x^2 - 1) \% n = 0$ or $(x-1)(x+1) \% n = 0$.
Then for n to be prime either $x \% n = 1$ or $x \% n = -1$.

So from these points we conclude:-

For n to be Prime,

$a^d \% n = 1$
or
$a^{d * 2^i} \% n = -1$
for some i, where $0 <= i <= s-1$

→ Code :-

```
int main() {
        int k=4;
        for (int n=1; n<100; n++)
            if ( isPrime(n,k))
                cout << n << " ";
        return 0;
    }
bool isPrime (int n, int k)
{   if (n<=1 || n==4) return false;
    if (n<=3) return true;
    // Finding r such that n = 2^(d*r) +1 for some r≥1
    int d = n-1;
    while (d%2 ==0)
        d/=2;
    for( int i=0; i<k; i++)
        if(millerTest (d,n))
            return false;
    return true;
}
bool millerTest( int d, int n) {
    int a = 2 + rand() % n-4;
    int x = pow (a, d) % n
    if (x==1 || x==n-1) return true;
    while (d != n-1)
    { x = (x * x) % n;
      d = d * 2;
      if (x==1)    return false;
      if (x==n-1) return true;
    }
return false;}
```

→ Primality Certificate

→ Examining the situation where n is probably Prime

→ In this situation we provide a certificate that n is a prime.

⇒

Certificate:

→ $n \in \mathbb{N}$ is prime iff there exist an element $a \in \mathbb{Z}/n\mathbb{Z}$ such that $a^m = 1$ but $a^{m/q} \neq 1$ and $q \geq \sqrt{n}$ for some integer m and a prime factor q.

Proof → Suppose we find an element a. [which satisfy all cond^n above]

m → integer, q → a prime factor,

Now, n is prime bcoz if p is a factor of n,

then $\gcd(a^{m/q} - 1, p)$ divides $\gcd(a^{m/q-1}, n)$

$\neq 1$ in $\mathbb{Z}/p\mathbb{Z}$. $\neq 1$ $\mathbb{Z}/p\mathbb{Z}$

But this means q divides $p-1$.

Since $q \geq \sqrt{n}$ so it is not possible

Hence n is prime.

→ if n is not prime, n must have a prime factor smaller than $\sqrt{n}$.