

③ PRINT PRIME TILL (N)

Way 1 $O(n\sqrt{n})$ (OLD WAY)

```
public static void main (String[] args)
{
```

```
Scanner s = new Scanner(System.in);
```

```
int n = s.nextInt();
```

```
for (int x = 2; x ≤ n; x++) ] ② aur ② se agge ki
```

```
{ boolean isPrime = true; } marke chalo digit is prime
```

```
for (int div = 2; div × div ≤ x; div++)
```

```
{ if (x % div == 0)
```

```
{ isPrime = false;
break;
}
```

Agar (x) , (\sqrt{x}) tak kisi se bhi divide hogaya toh prime nahi

```
if (isPrime == true)
```

```
{ Syso(x);
}
```

Time complexity = $n \times \sqrt{n}$
outer loop $x \rightarrow 2$ to n
inner loop $div: 2 \rightarrow \sqrt{x}$

WAY 2

SIEVE OF ERATOSTHENES

★ IT IS THE ONE OF THE MOST EFFICIENT WAYS TO FIND ALL PRIMES SMALLER THAN (n) , WHEN (n) IS SMALLER THAN (10 MILLION).

∴ COMPLEXITY OF SIEVE OF ERATOSTHENES = $O(n \log \log n)$ (WORST CASE)

[Way 1] OLD WAY

$n = 2500$

$\sqrt{n} = 50$

∴ $n \times \sqrt{n} = 2500 \times 50$
(Time complexity)

[WAY 2] SIEVE OF ERATOSTHENES ★

$n = 2500$

$\log n \approx 18$ ∴ $2^{18} = 2560$

$\log \log n \approx 4$

$2^4 = 16$

Time complexity = $n(\log \log n)$
 $= 2500 \times 4$

★ VERY HUGE DIFFERENCE

```
public static void main (String [] args) {
```

```
Scanner s = new Scanner (System.in);
```

```
int n = s.nextInt();
```

```
boolean [] arr = new boolean [n+1];
```

```
for (int i = 2; i * i < arr.length; i++)
```

Humne [n+1] size ka boolean array bana diya!

√n tak jayega loop!

```
{
    if (arr[i] == false)
```

```
{
    for (int j = i; i * j < arr.length; j++)
```

```
{
    arr[i * j] = true;
```

```
}
```

```
}
```

```
}
```

```
for (int i = 2; i < arr.length; i++)
```

```
{
    if (arr[i] == false)
```

```
{
    System.out.println(i);
```

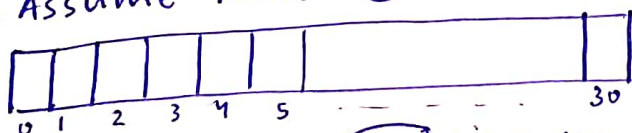
```
}
```

```
}
```

```
}
```

● DRY RUN

Assume that $n = 30$



Array Banega $(n+1)$ size ka kuki 0 se n tak aur $(n+1)$ elements hoty hai Aur $0-n$ Tak hum prime banaty hai!

Divisors (i) Hongey!

2 se \sqrt{n} Tak [Bas vo divisor hongy
2 se $\sqrt{30}$ Tak
2 se 5 tak jo khud prime hongy]

∴ 2, 3, 5

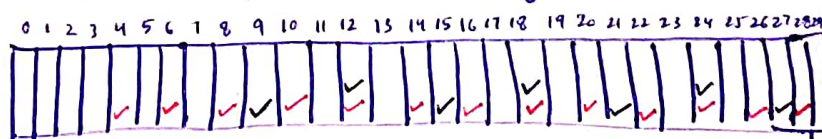
ki Table se hi hongy!

$\sqrt{30} \approx 5$

⇒ [0 aur 1] se Divisor (i) suru nahi hua kuki 0 aur 1 non-prime hai

⇒ PRIME NUMBERS are the numbers that are divisible by 2 Number

⇒ PRIME NUMBERS have 2 distinct Factors (1 and itself)



⇒ 2 ke Multiple [2x2] se start hongy!

⇒ 3 ke Multiple [3x3] se start hongy

⇒ 4 ke liye chalega nahi kuki vo (Non-prime == true).

⇒ 5 ke Multiple [5x5] se start hongy!

⇒ 6 ke liye LOOP chalega nahi becoz \sqrt{n} ($\sqrt{30} = 5$) tak hi consider hoga!

⇒ Aab (0, 1) ke alawa, jo bhi no. unchecked rah gaye vo sab prime no. hai

2, 3, 7, 11, 13, 17, 19, 23, 29

Time complexity : $O(n(\log(\log n)))$



$$O(n \log^* n)$$