Prime No

- Naive Approach
- Sqrt Approach
- Seive of Eronostuenes
- segmented Seive

Naive Approach

I/P → N

O/P → Prime / not

N < 10

(2 . . . . . N-1)

1, 2, ... , 9, 10

G ?

(N %. i) == 0 ??

↳ d

else

↳ Prime

N → Exactly / only

↓

2 factoor

↓,

1, 2

{ 1 is not prime }

```cpp
class Solution {
public:
    bool checkPrime(int n){
        for(int i = 2; i < n; i++){
            if(n%i == 0){
                return false;
            }
        }
        return true;
    }

    int countPrimes(int n) {
        int count = 0;
        for(int i = 2; i < n; i++){
            if(checkPrime(i)) count++;
        }
        return count;
    }
};
```

Seive of Eratosthenes

Given → n → return total no. of
Primes < N

N = 15

So we mark the true on true
numbers from 1 ——→ 15

But we don't need 1 & N so we mark
false

f
✗ T T T T T T T T T T
1 2 3 4 5 6 7 8 9 10 11

T   T   T   ✗ f
12   13   14   15

we know that the Smallest prime no → 2

if 2 → Prime
       ↓
    is it possible → all the multiple of 2
              also prime   → α

So → we mark the false all the
    multiple   of 2   upto N .

f	T	T	f	T	f	T	f	T	f	T
1	2	3	4	5	6	7	8	9	10	11

f	T	f	f
12	13	14	15

Same for ⟶ $\underline{\underline{3}}$

f	T	T	f	T	f	T	f	f	f	T
1	2	3	4	5	6	7	8	9	10	11

f	T	f	f
12	13	14	15

Same for $\underline{\underline{5}}$

f	T	T	f	T	f	T	f	f	f	T
1	2	3	4	5	6	7	8	9	10	11

f	T	f	f
12	13	14	15

Same for 7 , 11 , 13

So the prime are ⟶ 2, 3, 5, 7, 11, 13

⟹ ⑥

Algo

1) $2 \rightarrow N-1$,   [array no's] $\rightarrow$ Scive

mark all of them os
prime $\rightarrow$ true

2) Start from 2 till end, marks
all the no. comes in table 2
of non – prime

3) Repeat (2) till (N-1) only the
prime no.

4) Rest elements marked as prime
will be counted

5) Return Count

```cpp
class Solution {
public:
    int countPrimes(int n) {
        if(n == 0) return 0;

        // make a vector with mark all prime with true
        vector<bool> prime(n, true);

        // mark 0 & 1 -> false - bcoz it will not count in prime
        prime[0] = prime[1] = false;

        int count = 0;

        for(int i = 2; i < n; i++){
            // check in array if i is prime mark or not
            // if prime -> count++ -> and start the table of i -> mark all non-prime
            if(prime[i]){
                count++;

                // get the table of 2 -> mark false;
                for(int j = 2*i; j < n; j = j+i){
                    prime[j] = false;
                }
            }
        }
        return count;
    }
};
```

$T.C \longrightarrow$ outer loop $\Rightarrow n$

inner $\Rightarrow n \left[\dfrac{n}{2} + \dfrac{n}{3} + \dfrac{n}{5} \ldots \ldots \right]$

for prime

$n \left(\dfrac{1}{2} + \dfrac{1}{3} + \dfrac{1}{5} + \dfrac{1}{7} \ldots \ldots \right) \Rightarrow$ H.P of prime no.

taylor series

$O \left(n \left[\log (\log n) \right] \right) \Rightarrow O \left(n \log (\log n) \right)$

Segmented Sieve