Todays Content:

- → IsPrime
- → Prime Seive
- → Count factors

- Oct 5th: Holi
- Oct 24th: —
- Oct 26th: —

IsPrime(): number with only 2 factors

N = 10 : Not prime

N = 7 : Yes

IsPrime(): Count of factors of N = 2

↳ i = 1 ... n & count factors : TC : O(N)

↳ ✨ → i = 1 ... √n & count factors : TC : O(√N)

1) Print all Primes from 1 - N :

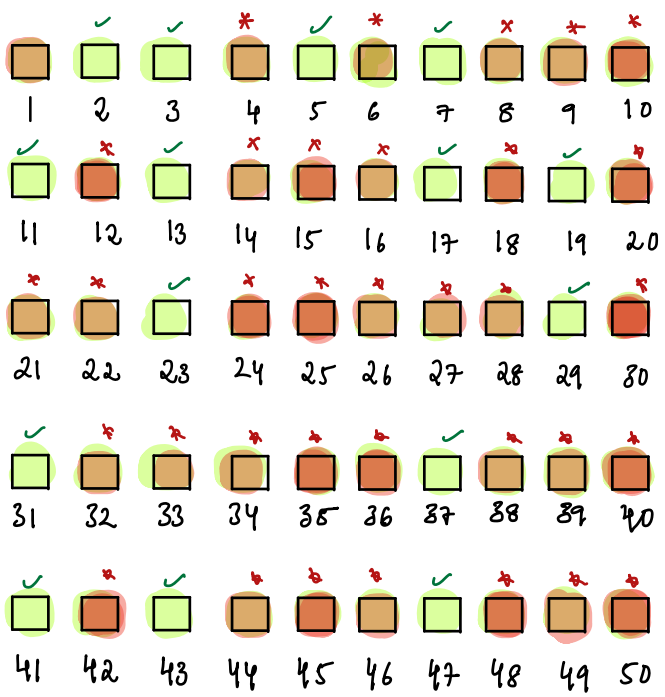
N = 10 : output : 2 3 5 7

Idea1 : Iterate on all numbers from 2 ... N & check if number is prime & print

TC : N * {√N} → O(N√N)

↳ TC for ease prime function

Idea2 : Say we need all prime 1-50



Pseudocode: Seive of Eratotheres / Prime Seive

```
void allPrime(int n){
    bool p[N+1] = T → { Initialize all value, without iterating }
                      TODO
    p[0] = F   p[1] = F
    i = 2; i<= N; i++){
        // We iterate on multiple of i, if i is a prime
        if (p[i] == T) { // i is a prime
            // iterate on multiple of i till N . . . . . .
            j = 2*i; j <= N; j = j + i){
                // j is mul of i & it is not prime
                p[j] = F
            }
        }
    }
    i = 2; i<= N; i++){
        if(p[i] == True) { print(i) }
    }
}
```

i
→ 2*i } +i
3*i } +i
4*i } +i
5*i
.
.

TC: Given N:

i	j : mul of i till N	Itera
2	mul of 2 till N	N/2
3	mul of 3 till N	N/3
4	—	
5	mul of 5 till N	N/5
6	—	
7	mul of 7 till N	N/7
:		
N	mul of N till N	N/N

Total Iterations:

$S = \frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} + \cdots \frac{N}{N}$

$= N\left[1/2 + 1/3 + 1/5 + 1/7 + \cdots 1/N\right]$

Sum of reciproals of all prime till N $\approx \log_2(\log(N))$

TC: $N * [\log(\log N)]$ SC: $O(N)$

$N = 2^{32}$ $\log_2(N) = 32$

↳ $\log_2(\log N) = \log_2(32) = 5$

Optimization to above idea :

10:40 → 10:50

N = 36

1 mut of i, if is prime

2 → 2*2 2*3 2*4 2*5 2*6 2*7

3 → 3*2 3*3 3*4 3*5 3*6 3*7

4 → *

5 → 5*2 5*3 5*4 5*5 5*6 5*7

2 3 2

6 → *

7 → 7*2 7*3 7*4 7*5 7*6 7*7

2 3 2 5 2

void allPrime(int n){ TC: O(N log(log N)) SC: O(N)

```
bool p[N+1] = T → { Initialize all value, without iterations }
                    TODO
p[0] = F  p[1] = F
i = 2; i*i <= N; i++) {                          opt1
    // We iterate on multiple of i, if i is a prime
    if (p[i] == T) { // i is a prime
        i = i*i; j <= N; j = j+i) {              opt2
            // j is mul of i & it is not prime
            p[j] = F
        }
    }
}
i = 2; i <= N; i++) {
    if (p[i] == True) { print(i) }
}
}
```

[Claim: last i value for which we will enter inner loop: $\sqrt{N}$]

i : [2 N] j = [i*i → N]

2

⋮

$\sqrt{N}$ → j = [N → N] 1 iteration

$\sqrt{N}+1$ → j = i*i > N : 0 iteration

$j = [\sqrt{N}+1]^2 = N+1+2\sqrt{N} > N$

$\sqrt{N}+2$ → no iterations

Find no: of factrs for all [1-N]

N=10:	1	2	3	4	5	6	7	8	9	10
#fats :	1	2	2	3	2	4	2	4	3	4

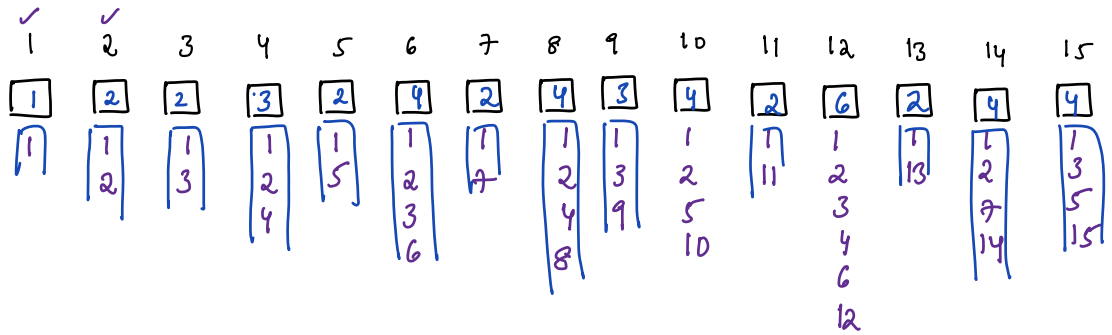
Idea1: for all numbers from 1-N iterate & calculate no: of factrs

TC: N√N

O(N) O(√N)

Idea2:

N=15.




```
int[] All factrs(int n){
    int fct[N+1] = {1}
    i=2; i<=N; i++){
        // i is factr to all mul of i = i 2i 3i 4i 5i....
        j=i; j<=N; j=j+i){
            // i is factr of j
            fct[j] = fct[j]+1      fct[j] = fct[j]+i
                                   sum of all factrs of j
        }
    }
    return fct[]
}
```

Dryrun: N=8

	1	2	3	4	5	6	7	8
fct[]	1	2	2	3	2	4	2	4

Table Given N

i	j : mul of i till N	iter
2	mul of 2 till N	N/2
3	mul of 3 till N	N/3
4	mul of 4 till N	N/4
5	mul of 5 till N	N/5
⋮		
N	mul of N till N	N/N

Total Iter =

$$S = \frac{N}{2} + \frac{N}{3} + \frac{N}{4} + \frac{N}{5} + \cdots \frac{N}{N}$$

$$N\left[\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots 1/N\right]$$

Sum of reciprocals of all natural numbers [2 N]

$\approx \log_2 N$

$$S = N * \log_2 N \Rightarrow \text{TC: } O(N \log_2 N)$$