Prime Numbers → +ve numbers with exactly 2 factors.
(1 & N)

Eg → 2, 3, 5, 7 . . .

-1 ↔ 1 } not prime factors

Q → check if the given integer is prime.

N = 9 Ans = false

13 Ans = true

Sol → # factors (N) = 2

30 → 1 * 30
2 * 15
3 * 10
5 * 6

```
if (N<=1) return false

for (i=2; i*i <= N; i++) {
    if (N%i==0) return false
}

return true
```

TC = O(√N) SC = O(1)

Q → Given an integer N, print all prime numbers from 1 to N.

N = 10 o/p → 2 3 5 7

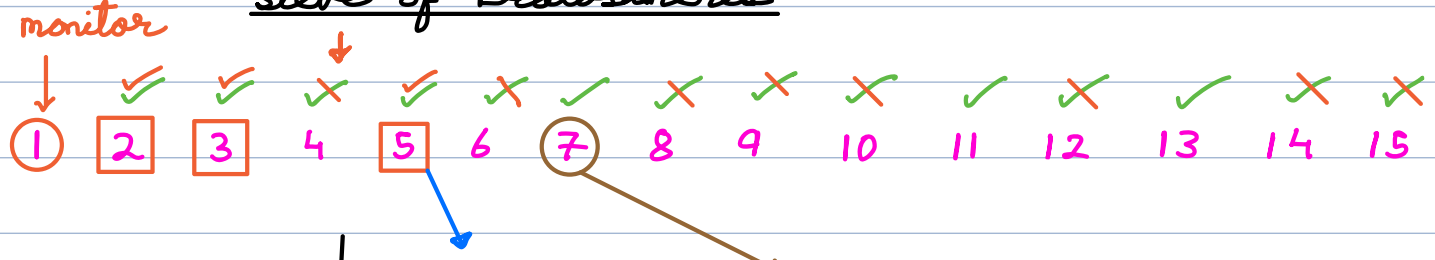
Bruteforce →

```
for i → 2 to N {
    if (checkPrime(i))
        print(i)
}
```

TC = O(N√N)

SC = O(1)

Sieve of Eratosthenes



roll No. → prime → chocolate

5 * 2 = 10

5 * 3 = 15

5 * (2 * 2) = 20

5 * 5 = 25 → first number 5 claim as false.

7 * 7 = 49

∀i, isP[i] = true

isP[1] = false isP[0] = false

```
for (i = 2; i*i <= N; i++) {
    if (isP[i]) {
        for (j = i*i; j <= N; j = j+i) {
            isP[j] = false
        }
    }
}
```

SC = O(N)

```
for i → 2 to N {
    if (isP[i]) print(i)
}
```

TC #iterations

2 → 4 6 8 . . . ~ N/2

3 → 9 12 15 . . . ~ N/3

4 → —

5 → 25 30 35 . . . ~ N/5

⋮

$\frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} \ldots < \frac{N}{2} + \frac{N}{3} + \frac{N}{4} + \ldots$

$= N\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{4} \ldots\right)$

$= N \boxed{\sum \frac{1}{i}}$ → $\int \frac{1}{x} dx = \log(x)$

TC = $O(N * \log(\log(N)))$ ✓ = $N \log(N)$

TC < $O(N \log(N))$ ✓

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

$\log_2(32) = 5$

Q → Given an integer N, find smallest prime factor (spf) for numbers from 2 to N.

N = 9

	2	3	4	5	6	7	8	9
spf →	2	3	2	5	2	7	2	3

Bruteforce → ∀i find spf.

TC = $O(N\sqrt{N})$

SC = $O(1)$

```
for i → 2 to √N {
    if ( N % i == 0) return i
```

```
}
return N
```

```
∀i, spf[i] = i
isP[1] = -1          isP[0] = -1
for (i = 2; i*i <= N; i++) {        i = ~~2~~ 3
    if (spf[i] == i) {  // is prime
        for (j = i*i; j <= N; j = j + i) {
            if (spf[j] == j)
                spf[j] = i
        }
    }
} return spf
```

spf[12] = ~~12~~ ~~2~~ 3 (x)

spf[]

SC = O(N)/O(1)

TC = O(N * log(log(N)))

Q → Given an integer N, find #factors for all numbers from 1 to N.

N = 9	1	2	3	4	5	6	7	8	9
#factors →	1	2	2	3	2	4	2	4	3

Bruteforce → ∀i, find #factors i.

TC = O(N √N) SC = O(1)

```
∀i cnt[i] = 0
for i → 1 to N {
    for (j = i; j <= N; j += i) {
```

```
        cnt[j]++   // i is factor of j
    }
}
return cnt
```

i	1	2	3	4	5	6
cnt →	0	0	0	0	0	0
	1	+1	+1	+1	+1	+1
		+1	+1	+1	+1	+1
				+1		+1
						+1
				3	2	4

SC = O(1)

TC = O(N log(N))

Prime Factorisation

$N = 40 \rightarrow 2^3 * 5^1$

$\rightarrow 2^0\ 2^1\ 2^2\ 2^3$

$\rightarrow 5^0\ 5^1$

$2^0 * 5^0 = 1$ $\quad 2^2 * 5^0 = 4$

$2^0 * 5^1 = 5$ $\quad 2^2 * 5^1 = 20$

$2^1 * 5^0 = 2$ $\quad 2^3 * 5^0 = 8$

$2^1 * 5^1 = 10$ $\quad 2^3 * 5^1 = 40$

2	40
2	20
2	10
5	5
	1

2^{0-3} and 5^{0-1}

$(3+1) * (1+1) = 4 * 2 = 8$

$N = 300 \rightarrow 2^2 * 3^1 * 5^2$

100

$4 * 3 * 25 = 300$

$2^{0-2} \rightarrow 2+1 = 3$

$3^{0-1} \rightarrow 1+1 = 2$

$5^{0-2} \rightarrow 2+1 = 3$

$3 * 2 * 3 = 18$ #factors

$N = p_1^a * p_2^b * p_3^c \ldots$

factors $= (a+1) * (b+1) * (c+1) \ldots$

$N = 20 \rightarrow 2^2 * 5$ $\quad (2+1) * (1+1) = 6$

spf

2	20
2	10
5	5
	1

```
ans = 1                          // pre-calculation ↑↑
u = spf[N]                       // smallest prime factor
while (u > 1) {                  // N > 1
    cnt = 0
    while (N % u == 0) {
        cnt++
        N /= u
    }
    ans *= (cnt + 1)
    u = spf[N]
}
return ans
```

TC = $O(\log(N))$

SC = $O(N)$

Multiple Queries

TC = $O(N \log(N) + Q \log(N))$