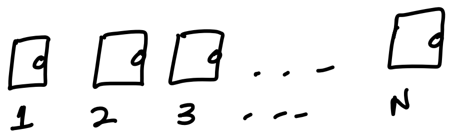
Q →



Initially all doors are closed.

A person comes & change the $\boxed{\text{state}}$ of door

open → close
close → open

like →
- 1, 2, 3 . . . N
- 2, 4, 6 - - - - -
- 3, 6, 9 - - - -
- 4, 8, 12 - - -

⋮

N

0 → close
1 → open

N=6   1   2   3   4   5   6

Find the # open doors in the end.

Ans = 2

time the state of door change → even → closed; odd → open

↳ = # factors of i for i^{th} door.

How many nos. from 1 to N have odd factors. ✓

10 → 1 * 10, 2 * 5 → 4 factors

20 → 1 * 20, 2 * 10, 4 * 5 → 6

25 → 1 * 25, $\boxed{5 * 5}$ → 3

36 → 1 * 36, 2 * 18, 3 * 12, 4 * 9, $\boxed{6 * 6}$ → 9

$N = x * x \Rightarrow x = \sqrt{N}$ is a integer

$\Rightarrow$ N is a perfect sq. ✓

How many perfect sq. we have from 1 to N. = $\sqrt{N}$

$\forall x <= \sqrt{N} \Rightarrow \boxed{x^2 <= N}$

Ans = $\lfloor \text{sqrt}(N) \rfloor$ ✓

$N = 30 \qquad \sqrt{N} = 5.47$

$1, 2, 3, 4, 5 <= \sqrt{N}$) sq

$\underbrace{1, 4, 9, 16, 25}_{\text{5 Ans}} <= N$

Prime No → Positive no. N s.t it has only 2 factors. [1, N].

2, 3, 5, 7, ...

-1

1 -1

Q→ Find # factors of N for given input N.

N=12 {1, 2, 3, 4, 6, 12} Ans = 6

N=10 {1, 2, 5, 10} Ans = 4

Bruteforce → ∀i st 1 <= i <= N check if i is factor of N.

(N % i == 0) ⇒ i is a factor of N

```
ans = 0;
for(i = 1; i <= N; i++){        // N -> N/2
    if(N % i == 0)  ans++;
} return ans;
```

TC = O(N) → ?

SC = O(1)

N = 16

1 * 16

2 * 8

4 * 4

```
ans = 0;
for(i = 1; i <= N/2; i++){
    if(N % i == 0)  ans++;
} return ans + 1;
```

1 2 3 4 5 6 7 8

✓ ✓ ✗ ✓ ✗ ✗ ✗ ✓

4 + 1 = 5

TC = O(N/2)

$\boxed{N = a * b}$ $\boxed{a <= b}$

Second largest factor can be = N/2

$\boxed{N = a * b \text{ s.t } a <= b}$

max possible value of a

N = 20   1 * 20
2 * 10
4 * 5
a   b

$\boxed{b = \frac{N}{a}} \Rightarrow a <= \frac{N}{a}$

$a^2 <= N \Rightarrow \boxed{a <= \sqrt{N}}$

a   b
$i \rightarrow \frac{N}{i}$

```
ans=1;
{
for(i=1; i*i<=N; i++){
    if(N%i == 0){
        if(i != N/i) ans += 2;
        else ans++;
    }
}
return ans;
```

ans → = 2 ⇒ prime

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

Q → Check if given no. N is prime?

prime $\boxed{N = 1 * N}$
7 → 1 * 7
10 → 1 * 10
2 * 5

N = 10   Ans = false
N = 7   Ans = true

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

$TC = O(\sqrt{N})$
$SC = O(1)$

Q → Find all prime numbers from 1 to N.

N = 6   1 2 3 4 5 6   Ans = {2, 3, 5}

N = 10   Ans {2, 3, 5, 7}

Bruteforce → ∀i st. $\underbrace{1 <= i <= N}_{O(N)}$, $\boxed{\text{check}}$ if i is prime no.
$O(\sqrt{N})$
$TC = O(N\sqrt{N})$

N = 17

1   2   3   4   5   6   7   8   9   10   11   12   13   14   15   16   17
×

10   15
20

25

↳ × not req.

2 * 7

Sieve of Eratosthenes

{2, 3, 5, 7, 11, 13, 17}

0 —— N

isPrime[N]

```
isPrime[N+1] = {true}   // ∀i isPrime[i] = true;
isPrime[0] = isPrime[1] = false;
for (i = 2; i*i <= N; i++) {
    if (isPrime[i] == false) continue;
    for (j = i*i; j <= N; j = j+i) {
        isPrime[j] = false;
    }
}
return isPrime;
```

$x \rightarrow \boxed{x^2}$

$2 * x$

$3 * x$

$\vdots$

$i = 5$

25 30 35 40 . —

$i = 3$

$\boxed{2} * 3 = 6$

$3 * 3 = 9$

$SC = O(N)$

$N = 10$

1	2	3	4	5	6	7	8	9	10
✗	✓	✓	✓	✓	✓	✓	✓	✓	✓

$TC \rightarrow \left(\frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} + \frac{N}{11} \dots\right)$ Σ(inner loops)

$i = 2$ 4, 6, 8, 10 . . . N $\left(\frac{N}{2}\text{ times}\right)$

$i = 3$ 9, 12, 15 . . . N $< \left(\frac{N}{3}\text{ times}\right)$

$i = 4$ ✗

$$< \underbrace{\left(\frac{N}{2} + \frac{N}{3} + \frac{N}{4} + \frac{N}{5} \dots \frac{N}{N}\right)}$$ (upperbound)

$$N\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots \frac{1}{N}\right) = N * \sum_{x=2}^{N} \frac{1}{x}$$

$$\approx N * \int_2^N \frac{1}{x}\,dx = N * [\log(N) - \log(2)] = N\log(N)$$

TC of sieve $= O(N\log(\log(N)))$ ≃ linear

$N = 2^{32}$ $\log_2(\log_2(2^{32})) = \log(32) = \log(2^5) = 5$

Q→ Find the count of all divisors/factors of all numbers from 1 to N.

N = 6   1 2 3 4 5 6

#factors → 1 2 2 3 2 4 → Ans ✓

Ans = 1+2+2+3+2+4 ✓

Brute → ∀i s.t $\underbrace{1 <= i <= N}_{O(N)}$ $\boxed{\text{count \# factors}}$ of i. $\quad TC = O(N\sqrt{N})$ ✓

$O(\sqrt{N})$

N = 12

1 2 3 4 5 6 7 8 9 10 11 12

#factors → 1

~~1~~ 2

~~1~~ 2 3

~~1~~ ~~2~~ 2 4 3

...

Ans → 1 2 2 3 2 4 2 4 3 4 2 6 ✓

1 2 3 4 5 6 7 8 9 10 11 12

```
o/p ✓
fact[i] = 0 ; ∀i
for (i=1 ; i <= N ; i++) {
    for (j = i ; j <= N ; j = j+i) {
        fact[j]++;
    }
}
return fact;
```

SC = O(1)

TC = O(N log(N)) ✓

i = 1   N times

i = 2   N/2 times

i = 3   N/3 times

$$TC = \sum_{i=1}^{N} N\left(\frac{1}{i}\right) = N \sum_{i=1}^{N} \frac{1}{i} = N\log(N) ✓$$

Q → Find all unique prime factors for all numbers from 1 to N.

N = 10

	1	2	3	4	5	6	7	8	9	10
Ans →	{}	{2}	{3}	{2}	{5}	{2, 3}	{7}	{2}	{3}	{2, 5}

```
for (i = 2 ; i <= N ; i++) {            // isPrime[i] = false;
    if (!pf[i].isEmpty()) continue;
    for (j = i ; j <= N ; j = j + i) {
        pf[j].add(i);
    }
}
return pf;
```

i = 2
j → 2 4 6 8 ---

i = 3
j → 3 6 9 12 --

i = 5
j → 5 10 15 ---

TC = O(N log(N))

SC = O(1)

N = 6

i = ~~2~~ ~~3~~ ~~4~~ ~~5~~ 6

pf[i]

1 → {}
2 → {2}
3 → {3}
4 → {2}
5 → {5}
6 → {2, 3}

Ans