Prime Numbers

10 | 1 × 10
2 × 5
5 × 2
10 × 1

$20 = 4 \times 5$ (2×10)
$24 = 4 \times 6$

Topics
- Count Factors
- P1 – Sieve
- P2 – # Factors 1–n
- P3 – smallest prime factor 1–n
- P4 – # Factors n
- P5 – open doors.

Q1
TC: $O(\sqrt{n})$
SC: $O(1)$

$n \to [1, n]$
Count of factors are even except perfect square

```
int countFactors(int n){
    ans=0
    for(int i=1; i<=√n ; i++){          // √n
        if(n % i == 0){
            if(i == n/i)
                ans+=1
            else
                ans+=2
        }
    }
    ret ans
}
```

ret countFactors(n)==2

Q2

Prime numbers:

Positive

a number that is only divisable by 1 and itself.

~~1~~, 2, 3, 4, 5, 6, 7, 8, 9, 10

~~1~~	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1
	2	3	4	5	6	7	8	9	10
			2		2		2	3	2
					3		4		5

P1 Given an integer n, check every number from 1 to n if it is a prime number. → output is an array bool

ex n=10

	1	2	3	4	5	6	7	8	9	10
F	F									
0	1	2	3	4	5	6	7	8	9	10

→ prime output

idea1

```
output[n+1] // init to false
for i=1 to n
  if(countFactors(i)==2)
     output[i]=true
  else output[i]=false
```

TC: $O(n \times \sqrt{n})$
SC: $O(1)$

idea 2
sieve of Erotosthen

bool

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18

<3 2×3
3×3
4×3
⋮

3x3

$\boxed{\text{spf}}$

important: strart from i*i not 1*i

optimized: smaller multiples already checkecked

when to use sieve?

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18

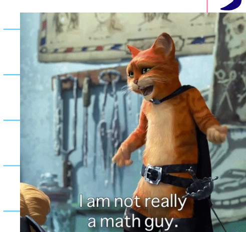
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18

SC: $O(n)$
or if ret isP $O(1)$

```
bool[] isPrimeN(int n){
    bool isP[n+1]; // all true
    init (isP, true); //O(n)
    isP[0] = isP[1] = false;
    for(i=2; i*i<=n; i++){
        if(isP[i]==true){ // is prime
            for(j=i*i; j<=n; j+=i){
                isP[j]= false
            }
        }
    }
}
```

$j <= \frac{n}{2}$ → optional assignment

TC: $n(\log(\log(n)))$



i	j ($= i \times i \to n$)	# iteration
2	4, 6, 8, 10, ..., n	$n/2$ +
3	9, 12, 15, ...	$n/3$ +
4	0	0 +
5	25, 30, 35, ...	$n/5$ +
6	0	0 +
⋮		⋮ +
$\sqrt{n}$		1

upper bound

iter: $\frac{n}{2} + \frac{n}{3} + \frac{n}{5} + \frac{n}{7} + \frac{n}{11} + \cdots$

do not worry about this detail. remember TC

$$= n\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \cdots\right) \xrightarrow{<=} \underbrace{n \sum_{i=2}^{\sqrt{n}} \frac{1}{i} < n \sum_{i=2}^{n} \frac{1}{i} <= n \int_2^n \frac{1}{x}\,dx}$$

$= n\log(n)$

TC <=

$TC <= n(\log(\log(n))) + n$

P2 Given an integer n, For all numbers from 1 to n

Count the factors of all those numbers

ex $n=8$

$$\begin{array}{cccccccc} 1 & 2 & 2 & 3 & 2 & 4 & 2 & 4 \\ 1, & 2, & 3, & 4, & 5, & 6, & 7, & 8 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ & & & 2 & & 2 & & 2 \\ & & & & & 3 & & 4 \end{array} \leftarrow \text{ans}$$

idea1 for $i \rightarrow 1$ to n

countFactors(i)

TC: $O(n\sqrt{n})$

idea2

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18

init

```
int[] countFactorsN( int n){
    count [n] = 0
    for( i=1; i<=n; i++){
        for( j=i ; j<=n ; j+=i ){
            count [j] += 1
        }
    }
    ret count
}
```

Q3

TC: $O(n \times \log n)$

SC: $O(1)$

i	j	iter
1	1, 2, 3, 4, ..., n	n +
2	2, 4, 6, 8, ...	$n/2$ +
3	3, 6, 9, ...	$n/3$ +
4	4, 8, 12, 16, ...	$n/4$ +
⋮		
n		

$$\#\text{iter}: n + \frac{n}{2} + \frac{n}{3} + \cdots + \frac{n}{n}$$

$$= n\left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right)$$

$$= n\left(\sum_{i=1}^{n} \frac{1}{i}\right) <= n \int_1^n \frac{1}{x}\,dx$$

TC: $O(n \times \log n)$

P3 Given an integer n, Find the smallest prime factor for all numbers SPF 2x7

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SPF	1	2	3	2	5	2	7	2	3	2	11	2	13	2	3	2	17	2

TC: $O(n \log(\log(n)))$
SC: $O(1)$

```
int[] smallestFactors(int n){
  spf[n+1]; // init to 0
  for(i=2; i*i<=n; i++){
    if(spf[i]==0){ // isPrime?
      spf[i]=i
      for(j=i*i; j<=n; j+=i){
        if (spf[j]==0){
          spf[j]=i
        }
      }
    }
  }
  for(i=2; i<=n; i++){
    if( spf[i]==0)   <- only for prime numbers assign the number itself.
      spf[i]=i
  }
  ret spf
}
```

in your assignment input is array of ints.

P4 Count the number factors of n, given the spf[i]

$1 \le i \le n$

ans		n
	Q4	45 → (1,45) (3,15) (9,5) → ans=6
	Q5	32 → (1,32) (2,16) (4,8) → ans=6

$45 = 3^2 \times 5$

$32 = 2^5$

$2^0 \quad 2^1 \quad 2^2 \quad 2^3 \quad 2^4 \quad 2^5$

3 | 0, 1, 2   0, 1 | 5

$3^0 \times 5^0 = 1$

$3^0 \times 5^1 = 5$

$3^1 \times 5^0 = 3$

$3^1 \times 5^1 = 15$

$3^2 \times 5^0 = 9$

$3^2 \times 5^1 = 45$

$$n = P_1^{n_1} \times P_2^{n_2} \times \cdots \times P_m^{n_m}$$

$$\boxed{\#\text{ of factors} = (n_1+1)(n_2+1)\cdots(n_m+1)}$$

ex

490	2
245	5
49	7
7	7
1	

$2^1 \times 5^1 \times 7^2$

$(1+1)(1+1)(2+1)$

$2 \times 2 \times 3$

12 ← ans

ex

600	2
300	2
150	2
75	3
25	5
5	5
1	

$2^3 \times 3^1 \times 5^2$

$(3+1)(2+1)(1+1)$

$4 \times 3 \times 2$

24

```
int countOfFactors(n){
    ans=1
    spf=smallestFactors(n)      → SC: O(n)
    ...                         TC: O(n log(log(n)))
    temp=n
    while(temp!=1){             O(log n)
        power=0                   ↓
        d= spf[temp]              HW
        while(temp!=1 && temp%d==0){
            power++
            temp = temp/d
        }
        ans = ans*(power+1)
    }
    ret ans
}
```

2^{4+1}

3^{1+1}

5×2

10

48	2
24	2
12	2
6	2
3	3
1	

open doors problem

ret floor(sqrt(n))

Quiz

$\boxed{\lfloor \sqrt{n} \rfloor}$

Number Of Open Doors

Problem Description

Given an integer **A**, which denotes the number of doors in a row numbered 1 to A. All the doors are closed initially.

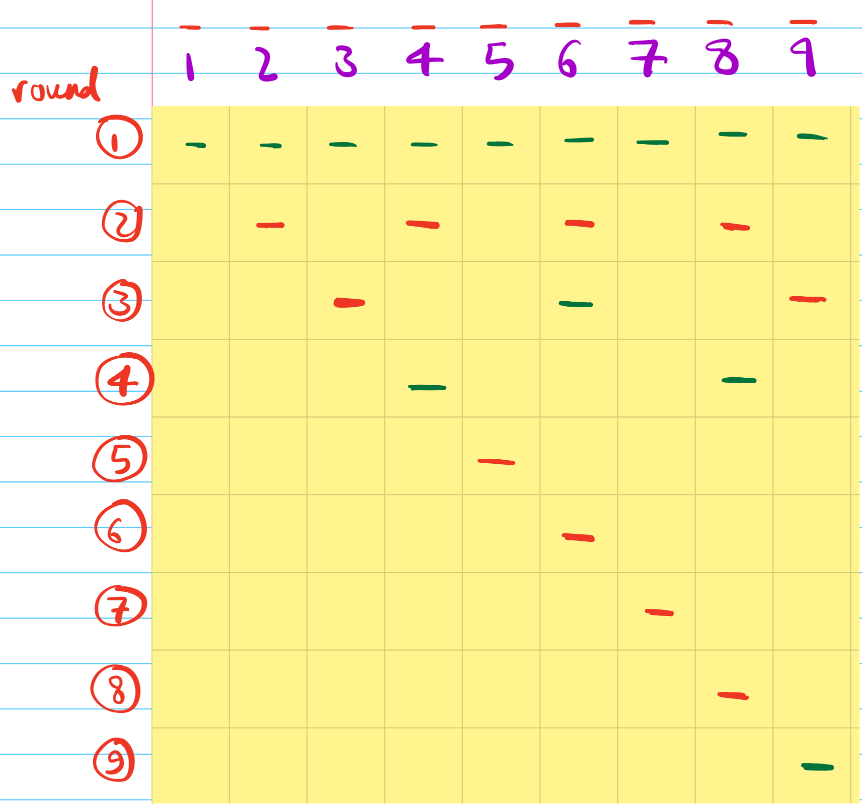
A person moves to and fro, changing the states of the doors as follows: the person opens a door that is already closed and closes a door that is already opened.

In the first go, he/she alters the states of doors numbered 1, 2, 3, ... , A.
In the second go, he/she alters the states of doors numbered 2, 4, 6 ....
In the third go, he/she alters the states of doors numbered 3, 6, 9 ...
This continues till the A'th go in, which you alter the state of the door numbered A.

Find and return the **number of open doors** at the end of the procedure.

Problem Constraints

$1 <= A <= 10^9$



– close
– open

ans = 3

$\sqrt{9}$