Today's agenda

- No. of factors
- Prime numbers
- Sum of N natural no.s
- floor & ceil
- Sqrt

Q) Count no. of factors

↳ Given a number N, Print the count of factors.

N = 24 → { 1 2 3 4 6 8 12 24 } → 8

N = 36 → { 1 2 3 4 6 9 12 18 36 } → 9

$$\begin{tikzcd} & 1 \\ N=36 \arrow[ur] \arrow[r] & 36(N) \end{tikzcd}$$

Brute force

```
P S V main( ) {
    Scanner scn = new Scanner(...);
        int n = scn.nextInt();

            int count = 0;
        for (int i = 1; i <= n; i++) {
            if (n % i == 0) {
                count ++;
            }
        }
        S.o.p (count);
}
```

iteration count ↓ N

Fact →

1 sec = 10^8 iterations

No. of iterations = N

$N = 10^9$

10^9 iterations

1×10^8 iterations = 1 sec

1 iteration: $\frac{1}{10^8}$ sec

10^9 iterations: $\frac{1}{10^8} \times 10^9$

= 10 sec

$\nearrow 10^{18}$ iterations

$N = 10^{18} \rightsquigarrow$ Seconds ?

10^8 iteration = 1 sec

1 iteration = $\frac{1}{10^8}$ sec

10^{18} iterations = $\frac{1}{10^8} \times 10^{18}$ = 10^{10} sec

10^{10} sec = 317.5 yrs

you $\longrightarrow$ child $\longrightarrow$ gc $\rightarrow$ 3rd $\rightarrow$ 4th 7th

→ optimal sol^n

//optimize

→ $j = N/i$

$i * j = N$ → i & j are 2 factors of N

↳ i & N/i are 2 factors of N

N = 24

i		N/i	count
1	<	24	+2
2	<	12	+2
3	<	8	+2
4	<	6	+2
6	>	4	
8	>	3	
12	>	2	
24	>	1	

$i <= N/i$

$i^2 <= N$

$i <= \sqrt{N}$

N = 36

i		N/i	Count = 0
1	<	36	2
2	<	18	4
3	<	12	6
4	<	9	8
6	= =	6	+1 → 9
9		4	
12		3	
18		2	
36		1	

```
int Count = 0;
for (int i = 1; i <= √n; i++) {
    if (n % i == 0) {
        Count = Count + 2;
    }
}
S.o.p (count);
```

// Psuedo code

```
P S V main ( ) {
        Scanner scn = new Scanner(...);
        int n = scn.nextInt();

        int Count = 0;
        for (int i = 1 ; i*i <= n ; i++) {
            if (n % i == 0) {
                if (i == n/i) { Count = Count + 1; }
                else { Count = Count + 2; }
            }
        }
        S.o.p (count);
}
```

iteration count

↓

$\sqrt{N}$

$\sqrt{10^{18}}$ iterations $= 10^9$ iterations

$N = 10^{18}$ ⤳ Seconds ?

1×10^8 iterations $= 1$ Sec

1 iteration: $\frac{1}{10^8}$ sec

10^9 iterations: $\frac{1}{10^8} \times 10^9$

$= 10$ Sec

Q) Prime numbers

↳ Given a number N, check if the number is a Prime no.

1 : Neither Prime nor composite.

Prime numbers ⟹ if the only factors are 1 and no. itself.

↓

Count of factors == 2

```
P s v main ( ) {
    Scanner scn = new Scanner(...);
    int n = scn.nextInt();

        int Count = 0;
        for (int i = 1 ; i*i <= n ; i++ ) {
            if (n % i == 0) {
                if (i == n/i) { Count = Count + 1; }
                else { Count = count + 2; }
            }
        }
        if (count == 2) { s.o.p ("Prime No."); }
            else { s.o.p ("Not Prime"); }
}
```

iteration count
↓
$\sqrt{N}$

Break till 10:38 PM

Quiz 1: Sum of all the numbers from 1 to 10.

Quiz 2: Sum of all numbers from 1 to 1000.
↳ 500500

Q) Sum of first N natural numbers.

→ Gauss (4th class)

$S = 1 + 2 + 3 \ldots \ldots 998 + 999 + 1000$
$S = 1000 + 999 + 998 \ldots \ldots 3 + 2 + 1$

$2S = 1001 + 1001 + 1001 \ldots \ldots 1001 + 1001 + 1001$

$2S = 1001 * 1000$

$S = \dfrac{1001 * 1000}{2} = \dfrac{1001000}{2} = 500500$

"Sum of first N naturals now

$S = 1 + 2 + 3 \ldots \ldots + (N-2) + (N-1) + N$
$S = N + (N-1) + (N-2) \ldots \ldots 3 + 2 + 1$

$2S = (N+1) + (N+1) + (N+1) \ldots \ldots (N+1) + (N+1) + (N+1)$

$2S = (N+1) * N \Rightarrow S = \dfrac{N * (N+1)}{2}$

Quiz3: Sum of 1st N whole numbers.

first 5 whole no.s → 0 1 2 3 4

first N whole no.s → $\cancel{0} + 1 + 2 + \cdots + N-1$

Sum of 1st N whole nos == Sum of first N-1 natural no.s,

$$\frac{N*(N+1)}{2} \rightarrow \frac{(N-1)*(N-1+1)}{2} \Rightarrow \frac{(N-1)N}{2}$$

floor(num) → just smaller or equal integer

Ex:	7.4	→ 7
	8.9	→ 8
	100.01	→ 100
	90	→ 90
	20.99	→ 20
	3	→ 3

Math.floor(num);

ceil(num) → just greater or equal integer

Ex: 7.4 → 8

8.9 → 9

100.01 → 101

90 → 90

20.99 → 21

3 → 3

math.ceil(num);

Q) Given N, return floor(sqrt(N))

ex: N = 60 → 7.... → 7
N = 31 → 5... → 5
N = 29 → 5.... → 5
N = 16 → 4.0 → 4

$6^2 \rightarrow 36$ $7^2 \rightarrow 49$ $8^2 \rightarrow 64$

```
public
static int sqrt (int N){
    int ans=1;
    int i=1;
    while(i*i<=n){
        ans = i;
        i++;
    }
    return ans.
}
```

N = 60

i	i*i<=N	ans
1	T	1
2	T	2
3	T	3
4	T	4
5	T	5
6	T	6
7	T	$\boxed{7}$
8	F ↳ exit	

↳ no. of iterations → $\sqrt{N}$

Tracing

```
P s v main( ) {
    Scanner scn = new Scanner(...);
    int n = scn.nextInt();

        int Count = 0;
    for (int i = 1 ; i*i<=n ; i++){
        if (n%i ==0){
            if(i== n/i){ Count = Count+1;}
              else { Count = Count+2;}
        }
    }
    s.o.p (count);
}
```

iteration Count
↓
$\sqrt{n}$

N=49

i	n%i==0	N/i	Count =0
1	t	49	2
2	x		
3	x		
4	x		
5	x		
6	x		
7	t	7	3
8			
9			
n			