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

N: 36 → 1

N: 36 → 36 (N)

Brute force

```
P S V main () {
    Scanner scn = new Scanner (system.in);
    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

10^8 iteration : 1 sec

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

10^9 iteration : $\frac{1}{\cancel{10^8}} \times \overset{10}{\cancel{10^9}}$

= 10 sec

10^{18} iteration

N = 10^{18} → time??

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 → child → gc → 3rd → 4th → 5th.....

8th yrs

→ better than Prev solⁿ

//optimize

$i*j = N$

↓

if you found out i, How will you find j corresponding to i ?!

↳ $j = N/i$

↳ factor always exists in Pair $\{i, N/i\}$

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	

$i <= N/i$

OR

$i^2 <= N$

N=36

i		N/i	
1	<	36	
2	<	18	
3	<	12	
4	<	9	
6	==	6	+1

//Psuedo code

```
P s v main() {
    Scanner scn = new Scanner(System.in);
        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);
}
```

$i*i<=n$

$i^2<=n$

$i<=\sqrt{n}$

iteration count

$\sqrt{n}$ ←

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

↳

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.

Prime no: if the only factors of a number are 1 and itself.

1 : Neither Prime nor composite.

↓

Count of factors == 2 → No. is a Prime no.
else → No. is not Prime

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

        int count = 0;
    for (int i=1; i*i <= n; i++){
          if(n%i==0) {
√n ←         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"); }
}
```

Break till 10:45 Pm

Quiz 1: Sum of all the numbers from 1 to 10.
↳ 55

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

Q) Sum of first N natural numbers.
↳ Gauss (4th class)

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

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

$1001 + 1001 + \ldots + 1001 + 1001 + 1001$

$2S = 1001 * 1000$

$$S = \frac{1001 * 1000}{2} = 500\,500$$

→ Sum of first N natural numbers

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

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

$(N+1) + (N+1) + (N+1) \ldots + (N+1) + (N+1) + (N+1)$

$2S = (N+1) * N$

$$\boxed{S = \frac{(N+1) * N}{2}}$$

Quiz3: Sum of 1^{st} N whole numbers.

first 5 whole no.s: $0+1+2+3+4$

first N whole no.s: $0+1+2+\dots+N-1$

$\cancel{0}+1+2+\dots+N-1 = 1+2+3+\dots+N-1$

↳ sum of first N-1 natural no.s!

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

floor(num) → just smaller or equal integer

Ex: 7.4 = 7

8.9 = 8

100.01 = 100

90 : 90

20.99 = 20

3 = 3

```
int n= 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

```
int n= Math. Ceil (num);
```

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

N>=1

ex: N=60 → 7.... → 7

N=31 → 5.x → 5

N=29 → 5.y → 5

N=16 → 4 → 4

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

        int ans=1;
        int i= 1;

        while (i*i<=n){
              ans = i;
              i++;
        }
        s.o.p (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	7
8	f ↳exit	

↳No. of iterations: $\sqrt{N}$