

```

476 |  

477 |   for (let num = 1; num <= 10; num++) {  

478 }|

```

- * In order to enter the for loop condition must be true / satisfied.
- * after the code is executed it will perform operation.
- * again try to enter for loop operation condition cycle repeats.
- * this keeps on running until the condition fails / becomes false.
 num ≤ 10 will become false when num = 11.

- ① initialization, num = 1
- ② num $\leq 10 \Rightarrow 1 \leq 10 \Rightarrow$ true
- ③ console.log("level 1");
- ④ num++ \Rightarrow num = 2
- ⑤ num $\leq 10 \Rightarrow 2 \leq 10 \Rightarrow$ true
- ⑥ console.log("level 2");
- ⑦ num++ \Rightarrow num = 3
- ⑧ num $\leq 10 \Rightarrow 3 \leq 10 \Rightarrow$ true
- ⑨ console.log("level 3");
- ⑩ num++ \Rightarrow num = 4
- ⑪ num $\leq 10 \Rightarrow 4 \leq 10 \Rightarrow$ true
- ⑫ console.log("level 4")

(operation)

initialization

condition

(operation)

initialization

condition

* Iterating in reverse

num > 0
(or)

```
482 | for (let num = 5; num >= 1; num--) {  
483 |   console.log(`level ${num}`);  
484 }
```

* Sum of 1st N natural numbers :

$$= \underline{1} + \underline{2} + \underline{3} + \underline{4} + \underline{5} + \dots + \underline{n}$$

$$\text{num} = 1 \times 2 \times 3 \times 4 \times 5 \times \dots \times n \cancel{\times (n+1)}$$

Eg: $n = 5$

$$\text{op: } 1 + 2 + 3 + 4 + 5 = 15$$

Eg: $n = 3$

$$\text{op: } 1 + 2 + 3 = 6$$

let sum = 0;

for (let num = 1; num <= n; num++) {

 sum = sum + num;

(a)

 sum += num;

}

sum = 0

/

%

*

10

15

20

25

30

35

40

n = 10

num = 1 \times 2 \times 3 \times 4 \times 5
 \times 6 \times 7 \times 8 \times 9 \times 10 $\cancel{\times 11}$

num = 4

$\rightarrow 4 < 5 \rightarrow \text{sum} = \text{sum} + \text{num}$

$\rightarrow \text{sum} = 0 + 4 = 4$

num = 5

$\rightarrow 5 < 5 \rightarrow \text{sum} = \text{sum} + \text{num}$

$\rightarrow \text{sum} = 4 + 5 = 15$

num = 6

$\rightarrow 6 < 5 \rightarrow \text{false}$

$n = 5$

num = 1, sum = 0

$\rightarrow 1 < 5 \rightarrow \text{sum} = \text{sum} + \text{num}$

$\rightarrow \text{sum} = 0 + 1 = 1$

num = 2

$\rightarrow 2 < 5 \rightarrow \text{sum} = \text{sum} + \text{num}$

$\rightarrow \text{sum} = 1 + 2 = 3$

num = 3

$\rightarrow 3 < 5 \rightarrow \text{sum} = \text{sum} + \text{num}$

$\rightarrow \text{sum} = 3 + 3 = 6$

num = 4

$\rightarrow 4 < 5 \rightarrow \text{sum} = \text{sum} + \text{num}$

$\rightarrow \text{sum} = 6 + 4 = 10$

* Efficient / optimal approach

What if $n = 10^{10}$ in our given problem? (you will get TLE ans)

→ for loop runs n times

→ 10^{10} times

→ 10^{10} instructions

10^8 instructions → 1s

$$10^{10} \text{ instructions} \rightarrow ? \cdot \frac{10^8}{10^8} = 100s$$

→ Our previous approach will not work

If $n > 10^8$. because it takes more than 1s.

* our for loop is running n times

In general, (online editor)

10^8 instructions → 1 second

→ your program is allowed to run for only 1s (TLE error)

$$\Rightarrow \frac{n(n+1)}{2} \quad (1 \text{ instruction only})$$

$$\Rightarrow n=5, \frac{5(5+1)}{2} = \frac{5 \times 6}{2} = 15$$

$$\Rightarrow n=10, \frac{10 \times 11}{2} = 55$$

* Factorial of a number ;

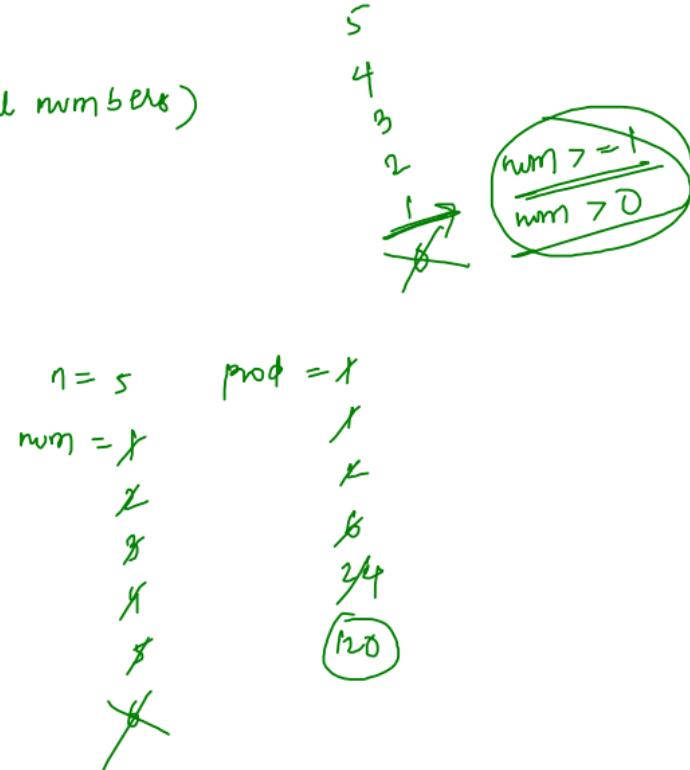
Eg: $n = 5 \Rightarrow 5!$ (product of first n natural numbers)

Op: $5 * 4 * 3 * 2 * 1 = 120$

 both are same

```
198 let prod = 1;  
199 for (let num = 1; num <= n; num++) {  
200   prod = prod * num;  
201 }  
202 console.log(prod);
```

$\text{num} > 0$
 (as)
for (let num = n; num $>= 1$; num--) {
 prod = prod * num;
}



* Even Sum :

Eg : $n = 12$

Op: $2 + 4 + 6 + 8 + 10 + 12 = 42$

init (operation) condition

let sum = 0;

for(let num = 2; num <= n; num += 2) {

 sum += num;

}

$n = 12$

$num = 2$

4

8

12

16

20

24

$sum = 0$

2

6

12

20

30

42

~~42~~

Using formula :

$$\frac{n(n+2)}{4}$$

$$\Rightarrow \frac{n(12+2)}{4}$$

$$\Rightarrow 42$$

$$\begin{aligned} \text{eg: } & (2+4+6+8+10 \\ & +12+14) \\ & n = 20 \quad +16+ \\ & \qquad\qquad\qquad 18+20) \end{aligned}$$

$$\begin{aligned} & \frac{20(20+2)}{4} \\ \Rightarrow & 5 \times 22 = 110 \end{aligned}$$

eg: $n = 13$

$$\frac{13(13+2)}{4}$$

$$\Rightarrow \frac{13+15}{4}$$

$$\Rightarrow 48.75$$

but when $n=13$, ans = 42

eg: $n = 21$

$$\frac{21(21+2)}{4}$$

$$\Rightarrow \frac{21+23}{4} = \frac{5+25 \times 2^3}{4} = \text{some fraction}$$

* When $n = \text{even}$

formula works fine,
but $n = \text{odd}$ we need
to handle this,

$n = 13 \}$ have same
 $n = 12 \}$ answer

Ily,
 $n = 21 \}$ have same
 $n = 20 \}$ answer

So do not calculate for
 $n = 13/21$ instead do
it for $12/20$ only

\Rightarrow if ($n \mod 2 != 0$) {
 $n = n - 1$

* break :

→ terminate / come out of current loop in execution whenever you see a break ;

* Continue ;

→ move to the next iteration of current loop in execution by ignoring all the code below it .

★ Check prime : (prime number is only divisible by itself and 1 but not any other numbers)

Eg : $n = 5$

Op : prime number

Eg : $n = 13$

Op : prime number

Q : Check 13 is a prime or not ?

[2, 12]

$$\times \boxed{13 \text{ } \mid \text{ } 2 = = 0} \quad 13 \text{ } \mid \text{ } 11 = = 0 \times$$

$$\times 13 \text{ } \mid \text{ } 3 = = 0 \quad 13 \text{ } \mid \text{ } 12 = = 0 \times$$

$$\times 13 \text{ } \mid \text{ } 4 = = 0$$

$$\times 13 \text{ } \mid \text{ } 5 = = 0$$

$$\times 13 \text{ } \mid \text{ } 6 = = 0$$

$$\times 13 \text{ } \mid \text{ } 7 = = 0$$

$$\times 13 \text{ } \mid \text{ } 8 = = 0$$

$$\times 13 \text{ } \mid \text{ } 9 = = 0$$

$$\times 13 \text{ } \mid \text{ } 10 = = 0$$

$\Rightarrow 13$ is not divisible
by any other number
except for 1 & 13

$\Rightarrow 13$ is a prime
number

⇒ converting to code,

we are checking $[2, n-1]$

```
for (let num = 2; num < n; num++) {  
    if ( $n \% num == 0$ ) {  
        console.log("Not a prime number");  
    } else {  
        console.log("prime number");  
    }  
}
```

* WRONG

Code

we need
to correct

n is divisible by num
⇒ It is violating prime
property

$$num = 2 \quad n = 15$$

$$15 \% 2 == 0 \times
'prime'$$

"deciding the answer
Inside loop is like deciding
the answer by looking at
a single num" (wrong)

* you should look at all the
num (ie all checks) before
giving out your statement as "prime"

let isPrime = true; (assume initially given 'N')
as prime

for (let num = 2 ; num < n ; num++) {

if (N % num == 0) {

isPrime = false;

break;

}

}

if (isPrime) {

console.log ("a prime number");

}

else {

console.log ("not a prime number");

}

* flag variable

Technique

(to store some findings / results)

violating prime property

}

* deciding the answer outside the loop is like deciding the answer after all the checker (looking at all num)