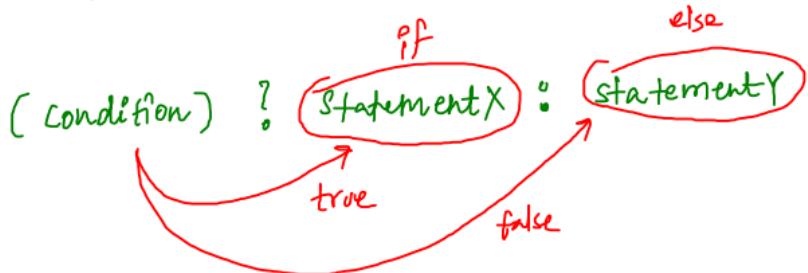


* Ternary operator :

(short hand if else)



```
378 // ternary operator
379 const age = 18;
380 if (age >= 21) {
381   console.log("You can drink alcohol");
382 } else {
383   console.log("You cannot drink alcohol");
384 }
385
386 const ans = age >= 21 ? "You can drink alcohol" : "You cannot drink alcohol";
387 console.log(ans);
388
389 age >= 21
390   ? console.log("you can drink alcohol")
391   : console.log("you cannot drink alcohol");
```

* Extra point :

→ Ternary operators are mostly used with conditional rendering of React Components.

* For Loop :

```
411 | for (let cnt = 1; cnt <= 10; cnt++) {  
412 |   console.log(`pushup ${cnt}`);  
413 | }
```

- ① initialisation, $\text{cnt} = 1$
- ② condition, $\text{cnt} \leq 10$, $1 \leq 10$? True
- ③ enter loop, op: pushup 1
- ④ operation, $\text{cnt}++$, $\text{cnt} = 2$
- ⑤ condition, $\text{cnt} \leq 10$, $2 \leq 10$? True
- ⑥ enter loop, op: pushup 2
- ⑦ operation, $\text{cnt}++$, $\text{cnt} = 3$
- ⑧ condition, $\text{cnt} \leq 10$, $3 \leq 10$? True
- ⑨ enter loop, op: pushup 3
- ⑩ operation, $\text{cnt}++$, $\text{cnt} = 4$
- ⑪ condition, $\text{cnt} \leq 10$, $4 \leq 10$? True
- ⑫ enter loop, op: pushup 4
- ⑬ operation, $\text{cnt}++$, $\text{cnt} = 5$
- ⑭ condition, $\text{cnt} \leq 10$, $5 \leq 10$? True
- ⑮ enter loop, op: pushup 5
- ⑯ $\text{cnt}++$, $\text{cnt} = 6$, $\text{cnt} \leq 10$, $6 \leq 10$, pushup 6
- ⑰ $\text{cnt}++$, $\text{cnt} = 7$, $\text{cnt} \leq 10$, $7 \leq 10$, pushup 7
- ⑱ $\text{cnt}++$, $\text{cnt} = 8$, $\text{cnt} \leq 10$, $8 \leq 10$, pushup 8
- ⑲ $\text{cnt}++$, $\text{cnt} = 9$, $\text{cnt} \leq 10$, $9 \leq 10$, pushup 9
- ⑳ $\text{cnt}++$, $\text{cnt} = 10$, $\text{cnt} \leq 10$, $10 \leq 10$, pushup 10
- ㉑ $\text{cnt}++$, $\text{cnt} = 11$, $11 \leq 10 \rightarrow F$ (Terminate)

- * In order to enter the loop always the condition must be true.
- * after the code inside for loop is executed it will perform operation
- * operation  condition cycle repeats.
- * when condition is false, terminate or come out of the for loop.
- * for loops are used to replace repetitive tasks.
- * If there is no condition or if the condition is always true that leads to an infinite loop.

$\text{cnt} = 0 \leftarrow 8 + 8 + 8 + 7 + 8 + 9 + 10 \quad (11) \times$

+1 +1 +1 (operation)

↓ Initialization

↓ $\text{cnt} \leq 10$ condition

$\text{cnt} = 8 \leftarrow 8 - 8 - 8 + 6 \times$

-1 -1 -1 (operation)

↓ Initialization

↓ $\text{cnt} > 0$ condition

Q: Sum of 1st N natural numbers :

$$n = 5, \text{ sum} = 0 \times 5 \neq 15$$

Eg: $n = 5$

Op: $1 + 2 + 3 + 4 + 5 = 15$

Eg: $n = 3$

Op: $1 + 2 + 3 = 6$

① Sum = 0

② Iterate for n times

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

Sum = Sum + num;

}

$$\text{num} = 1 \neq 8 \neq 4 \neq 6 \times$$

$$\text{sum} = 8 \times 5 \neq 15 \checkmark$$

① num = 1

② num <= 5, 1 <= 5

③ sum = sum + num, sum = 0 + 1 = 1

④ num++ , num = 2

⑤ num <= 5, 2 <= 5

⑥ sum = sum + num, sum = 1 + 2 = 3

⑦ num++ , num = 3, 3 <= 5, sum = 3 + 3 = 6

⑧ num++ , num = 4, 4 <= 5, sum = 6 + 4 = 10

⑨ num++ , num = 5, 5 <= 5, sum = 10 + 5 = 15

⑩ num++ , num = 6, 6 <= 5 X

Another approach :

$$\Rightarrow \frac{n(n+1)}{2}$$

$$= \frac{5(5+1)}{2}$$

$$= \frac{5 \times 6}{2} = 15$$

* 1st approach will not work for large N values,
e.g.: $N > 10^8$ leads to TLE

Instructions $\rightarrow n$ iterations/times

✓ 2nd approach $\rightarrow 1$ instruction
(Better than 1st)

* In general, your online IDE's can run at max 10^8 instructions (iterations)

if $10^8 \rightarrow 18$ (only 18 is given to your program)

$n = 10^9$ (not a possible TC by constraints)

for $\rightarrow 10^9$ times

(If your program exceeds 18 you will get TLE)

wk1, $10^8 \rightarrow 18$

$10^9 \rightarrow 10^8 \times \rightarrow$ TLE using 1st approach

① 10^8 iterations $\rightarrow 18$ (anywhere)

② your program is given only 18 time slot. If it exceeds this slot \rightarrow TLE

③ when will our program > 18 ? $> 10^8$ iterations \rightarrow TLE

const name = 'anurag'

console.log(`Hi I am \${name}`) → Hi I am name

console.log('Hi I am \${name}') → Hi I am name

console.log(`Hi I am \${name}`) → Hi I am anurag

* only used with string template

const sum = 15;

console.log(sum) → 15

Q: factorial of a number :

$$n = 5, \text{ fact} = 1$$

Ex: $n = 5$

Op: $n! \Rightarrow n \times n-1 \times n-2 \times \dots \times 1$

$5! \Rightarrow 5 \times 4 \times 3 \times 2 \times 1 = 120$

fact = 1 num = 5; num >= 1; num-- (reverse)

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

}

console.log(fact)

① num = 1, num <= 5 ($1 \leq 5$)

$$\text{fact} = 1 + 1 = 1$$

② num = 2, num <= 5 ($2 \leq 5$)

$$\text{fact} = 1 + 2 = 2$$

③ num = 3, 3 <= 5

$$\text{fact} = 2 + 3 = 6$$

④ num = 4, 4 <= 5

$$\text{fact} = 6 + 4 = 24$$

⑤ num = 5, 5 <= 5

$$\text{fact} = 24 + 5 = 120$$

⑥ num = 6, 6 <= 5 X

Q: even sum ;

$$n = 12$$

Eg: $n = 12$

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

let sum = 0;

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

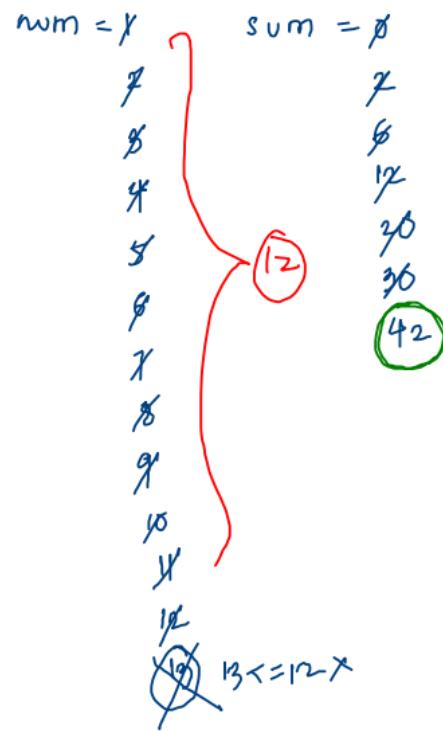
 if (num % 2 == 0) {

 sum = sum + num;

 }

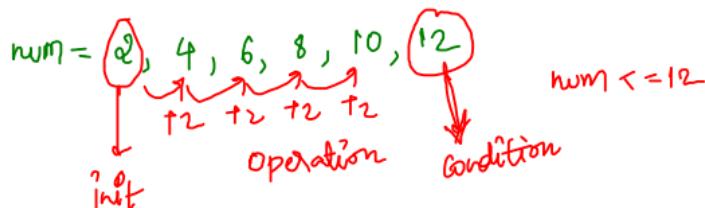
}

console.log(sum);



Another approach :

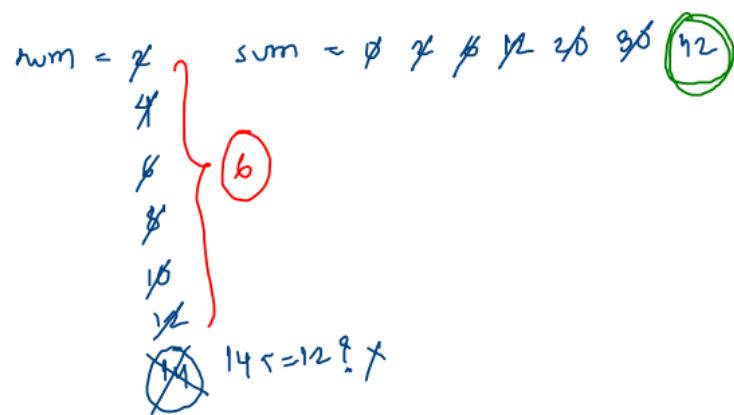
→ why do you want to iterate on all numbers,
iterate on what you require.



* this is better than previous approach
because 1st approach $\rightarrow n$ iterations
2nd approach $\rightarrow \frac{n}{2}$ iterations

```

let sum = 0;
      num = num + 2
      ↑
for(let num = 2; num <= n; num += 2) {
    sum = sum + num;
}
console.log(sum);
  
```



Formula : (efficient \rightarrow 1 instruction)

$$\frac{n(n+2)}{4} \Rightarrow \frac{\cancel{n}^3 + \cancel{n}^4}{\cancel{4}^1} = 42$$

let $n = 13$,

$$\frac{13 + 15}{4} = 48.75 \text{ (not as expected)}$$

$$n = 12 \rightarrow 42 \quad (2, 4, 6, 8, 10, 12)$$

$$n = 13 \rightarrow 42 \quad (2, 4, 6, 8, 10, 12)$$

* when n is odd, handle this formula carefully,

$n = 13$ is same as $n = 12$

? instead of calculating for 13, calculate for 12

if (n is odd) {

$n = n - 1;$
 }

 apply the formula

$$n = 20 \rightarrow \frac{20 + 22}{4} = 110$$

$$n = 21 \rightarrow \frac{21 + 23}{4} = \text{some unexpected value}$$

21 \leftarrow 20 \Rightarrow have same even sum

* break :

→ whenever you see this, Just get out of the current loop.

```
429 for (let num = 1; num <= 5; num++) {  
430   console.log(`You are at level ${num}`);  
431  
432   if (num == 3) {  
433     console.log("Sorry you cannot proceed to next level");  
434     break;  
435   }  
436 }  
437  
438 console.log("for loop completed");
```

① num = 1, 1 ≤ 5

→ "you are at level 1"

→ 1 == 3 ✗

② num = 2, 2 ≤ 5

→ "you are at level 2"

→ 2 == 3 ✗

③ num = 3, 3 ≤ 5

→ "you are at level 3"

→ 3 == 3 ✓

→ "Sorry....."

→ break; (terminate current loop)

④ "for loop completed"

* Continue :

→ whenever you see this, directly move to operation
by ignoring/skipping the entire code below it.
(Ignore current iteration and move to next iteration).

~~if (num == 3) {~~

```
441 for (let num = 1; num <= 5; num++) {  
442   if (num == 3) {  
443     console.log("Congratulations!!! you can skip this due to bonus points");  
444     continue;  
445   }  
446   console.log(`You are at level ${num}`);  
447 }  
448  
449 console.log("for loop completed");
```

① num = 1

→ 1 == 3 ✗

→ You are at level 1 ✓

② num = 2

→ 2 == 3 ✗

→ You are at level 2 ✗

③ num = 3

→ 3 == 3 ✓

→ "Congrats...---" ✓

→ Continue ()

④ num = 4

→ 4 == 3 ✗

→ You are at level 4 ✗

⑤ num = 5

→ 5 == 3 ✗

→ You are at level 5 ✗

* Quadrants (Doubt) :

The mystery room is divided into four chambers and each chamber will have two boxes storing balls.

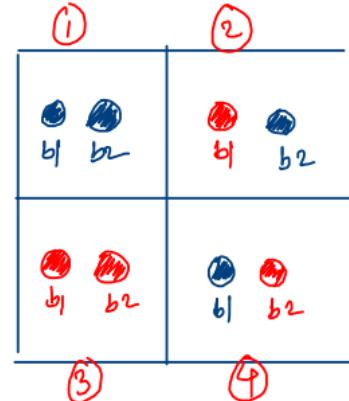
Now if both the boxes have blue balls, it must belong to chamber 1.

Similarly if the first box has red balls and second box has blue balls, they must belong to chamber 2.

If both the boxes have red balls, they must belong to chamber 3.

Finally if the first box has blue balls and second box has red balls, it must belong to chamber 4.

You are given number of balls in each box and if the number has a negative sign it means the balls are red else the balls are blue if the sign is positive.



c-①

both are blue

$$b_1 > 0, b_2 > 0$$

c-②

$$b_1 - \text{red}, b_2 - \text{blue}$$

$$b_1 < 0, b_2 > 0$$

c-③

both are red

$$b_1 < 0, b_2 < 0$$

$$\text{eg: } b_1 = 10 \quad b_2 = 6 \quad | \quad b_1 = 9 \quad b_2 = -13$$

⑩

⑥

c-①

⑨

⑬

c-②

-ve \rightarrow red

+ve \rightarrow blue