

## \* short circuiting:

A and B and C and D

T and T and T and T → True

① F and ~~T and T~~ and T → False

① T and ② F and ~~T and T~~ → False

① T and ② T and ③ F and ~~T~~ → False

① T and ② T and ③ T and ④ F → False

→ break when you see false

A or B or C or D

F or F or F or F → False

① T or ~~F or F or F~~ → True

① F or ② T or ~~F or F~~ → True

① F or ② F or ③ T or ~~T~~ → True

① F or ② F or ③ F or ④ T → True

→ break when you see true.

\* Leap Year :

what? (366 day/s) (Feb 29<sup>th</sup>)

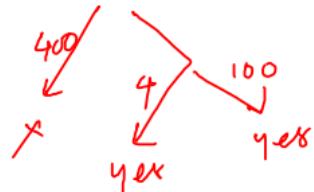
1. divisible by 400 (or)  $\rightarrow A$

2. divisible by 4 and not divisible by 100  $\rightarrow B$   
 { C } { D }

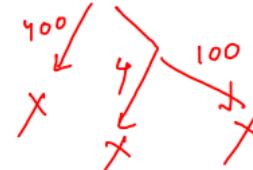
Eg: 2024  $\Rightarrow$  Leap Year



Eg: 5000  $\Rightarrow$  not Leap Year



Eg: 2050  $\Rightarrow$  not leap year



A or B

A or (C and D)

year % 400 == 0  $\rightarrow A$

year % 4 == 0  $\rightarrow C$

year % 100 != 0  $\rightarrow D$

A || (C & D)

\* Which case :

[‘A’ – ‘Z’] → 1 (capital alphabets)

[‘a’ – ‘z’] → 0 (small alphabets)

→ -1 (not any alphabet)

Eg:    x    |    f    |    @  
      → 1    |    → 0    |    -1

if  $\text{ch} \geq \text{A}$  and  $\text{ch} \leq \text{Z}$  → ch is a capital alphabet

else  
if  $\text{ch} \geq \text{a}$  and  $\text{ch} \leq \text{z}$  → ch is a small alphabet

else

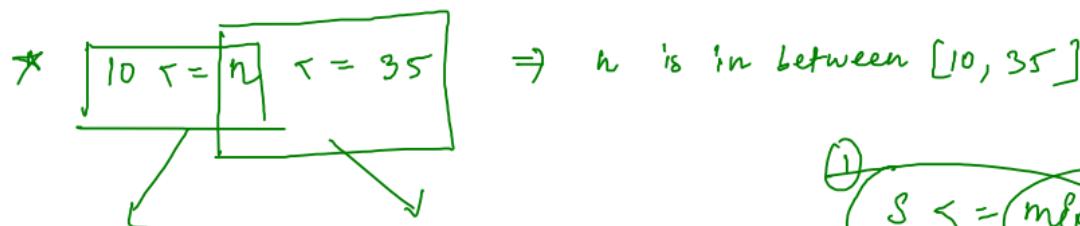
→ ch is not an alphabet

→ check whether the given number is in between [start, end] (inclusive)

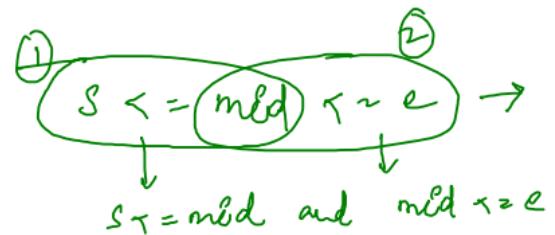
$$\text{num} = 20$$

$$[10, 35]$$

✓  $\downarrow$  true  
✗  $\downarrow$  false



```
if (start <= n && n <= end) {  
    console.log ("Yes it is in range");  
}  
else {  
    console.log ("Not in range");  
}
```



\* Big Light :

①  $g: 5m$      $s: 7m$   
 $v_1: 3m/s$      $v_2: 2m/s$

$t = 0$	5m, 7m	After 2 <sup>nd</sup>
$t = 1$	8m, 9m	Both g, s are at equal heights
$t = 2$	11m, 11m	op: true

②  $g: 5m$      $s: 7m$   
 $v_1: 2m/s$      $v_2: 3m/s$

$t=0$	5m, 7m $\rightarrow$ 2m	as long as you go,
$t=1$	7m, 10m $\rightarrow$ 3m	you will not meet
$t=2$	9m, 13m $\rightarrow$ 4m	op: false
$t=3$	11m, 16m $\rightarrow$ 5m	

- $h_1, h_2$   
 $v_1, v_2$
- ①  $t=0, h_1 == h_2 \rightarrow \text{true}$
  - ②  $h_1 < h_2$  and  $v_1 \neq v_2 \rightarrow \text{false}$
  - ③  $h_2 < h_1$  and  $v_2 \neq v_1 \rightarrow \text{false}$
  - ④ After some t' seconds,

$$h_1 = h_1 + tv_1$$

$$h_2 = h_2 + tv_2$$

Both heights should be same

$$h_1 + tv_1 = h_2 + tv_2 \quad h_1 = 5m$$

$$h_1 - h_2 = tv_2 - tv_1 \quad h_2 > 7m$$

$$h_1 - h_2 = t(v_2 - v_1) \quad v_1 = 10m/s$$

$$t = (h_1 - h_2) / (v_2 - v_1)$$

$$\boxed{t = 0.2}$$

not possible

\* t should be integer (+ve)

$$(h_1 - h_2) \% (v_2 - v_1) = 0$$

\*\*

1. understand

→ ip, op

2<sup>y</sup>. put your thoughts on paper

(take examples)

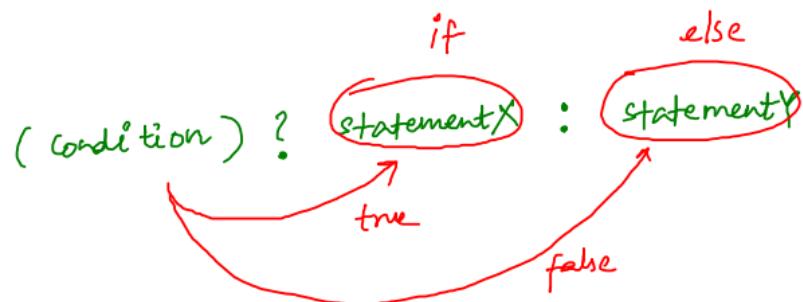
(Analyze)

3. write some steps and verify  
with the examples

4. code → errors (logic)  
(testcases)

## \* Ternary operator:

(short hand if else)



## # Recap:

### JS Fundamentals part - ①

#### 1. operator precedence

↳ Brackets (mdn Docs)

#### 2. conditionals

↳ if, else, else if

↳ ternary operator

↳ short circuiting

(avoid unnecessary) checker

#### 1. console.log

#### 2. variables and values (7/3, conventions, rules)

#### 3. let, const, ~~var~~ → functions

    (=, +=, -=, --)

#### 4. operators (arithmetic, assignment, comparison)

    (+, -, \*, /, \*\*, %)      (>, <, >=, <=)

    logical (||, !, !)      (==, !=, ==, !=)

#### 5. string templating \${ }

#### 6. type conversion and type coercion

Number ( )

String ( )

Boolean ( ) → Ⓛ

(S + N → S + N) → concat

and other operations (S) → N

(S - N → N - N)

DSA

datastructures and algorithms  
(language independent)

array

linked list

stack

queue

tree

graph

heaps

maps

sorting (in, bsb, sele)

search

dynamic programming

divide and conquer

recursion, backtracking

greedy

J8

problem solving (Basic)

1, 2 (coding)

3 - HTML, CSS (static)

4 - Adv JS (app)