★ logical operators : (AND, OR, NOT)

① AND – Both should satisfy

A: Sarah has a Driver's license → T / F

B: Sarah has good vision → T / F

② OR – atleast one of the statement must satisfy

★ A and B and C and D = True only when all stmts are True else False

★ A or B or C or D = False only when all stmts are False else True

A	B	A and B
T	T	T
T	F	F
F	T	F
F	F	F

A	B	A OR B
T	T	T
T	F	T
F	T	T
F	F	F

③ NOT - True → False
False → True

A	B	NOT A	NOT B
T	T	F	F
T	F	F	T
F	T	T	F
F	F	T	T

Question :

```
Const age = 16;
Const A = (age >= 20) ;   (16 >= 20) (False)
Const B = (age < 30);     (16 < 30) (True)
```

① ! A = ! false = true

② A and B = false AND True = false

③ A or B = false OR True = True

④ ! A and B = ! false AND True
= True AND True = True

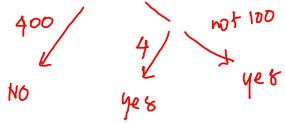
⑤ A or ! B = False OR ! true
= False OR False = False

★ Leap Year :

what ? (366 days) (Feb 29th)

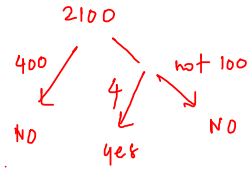
1. divisible by 400 → A
   (or)
2. divisible by 4 and not divisible by 100 → B
   C D

eg: 2024



⇒ Leap Year

eg: 2100



⇒ not a Leap Year

eg: 2050

⇒ not a Leap Year

A or B

A or (C and D)

A ⇒ year % 400 == 0

C ⇒ year % 4 == 0

D ⇒ year % 100 != 0

A || (C && D)

★ constraints : 1000 <= year <= 9999

⇒ all test cases will have input in range [1000, 9999]

⇒ what kind / what to expect from test cases

* which case :

'A' → capital → 1

'a' → small → 0

'#', → not alphabet
'@', '*' → -1

How do decide 'ch' is capital ?

'A' - 'Z' ⇒ [65, 90]

⇒ ASCII (given ch) should be in B/w 65 and 90.

⇒ ASCII (given ch) should be in B/w 97 and 122.

* Given a num check whether it is in the given range or not

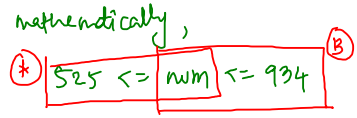
num = 12 → false

= 652 → true

[525, 934]

[start, end]

mathematically,



⇒ A AND B

```
if (525 <= num && num <= 934)
    cl ("yes In Range");
}
else {
    cl ("Not In Range");
}
```

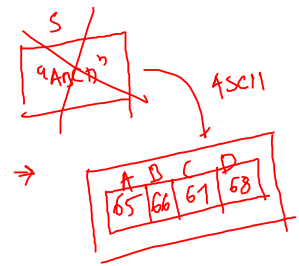
★ ASCII values :

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

(0 — 127) ASCII values

Const s = "ABCD"



"A" - "Z" "a" - "z"

[65, 90] [97, 122]

① String. fromCharCode (ascii)

⇒ (ASCII → characters)

② "A". charCodeAt (0)

→ (ch → ASCII)

* **Big Light :**

① h_1 h_2

q : 5m s : 7m

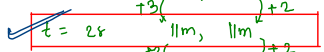
v_1 : 3m/s v_2 : 2m/s

time, $t = 0s$ 5m, 7m

$t = 1s$ 8m, 9m (+3, +2)

$t = 2s$ 11m, 11m (+3, +2)

$t = 3s$ 14m, 13m (+3, +2)



Qs : will they meet

⇒ Op : true

* $T = \frac{h_2 - h_1}{v_1 - v_2} = \frac{7-5}{3-2} = 2s$

* $T = \frac{7-5}{2-3} = \frac{2}{-1} = -2$

② q : 5m s : 7m

v_1 : 2m/s v_2 : 3m/s

time,				
$t = 0$	5m	7m	(7m − 5m = 2m)
$t = 1$	7m	10m	(10m − 7m = 3m)
$t = 2$	9m	13m	(13m − 9m = 4m)
$t = 3$	11m	16m	(5m)
$t = 4$	13m	19m	(6m)

⇒ Op : false

$T = h_2 - h_1 / v_1 - v_2$

h_1, h_2

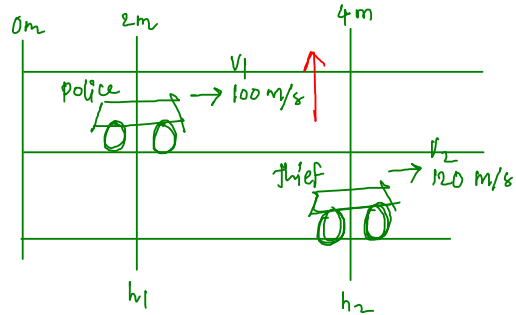
v_1, v_2

① $t = 0$, $h_1 == h_2$ → True

② $h_1 < h_2$, $v_1 <= v_2$ → False

③ $h_2 < h_1$, $v_2 <= v_1$ → False

} vice versa



⇒ Can police catch thief ? → thief is already ahead of police and he is moving faster than police

→ NO

police > 120 m/s

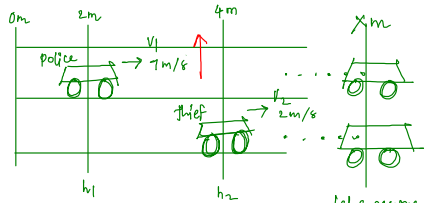
= 120 m/s (Can't catch)

④ $h_1 < h_2$, $v_1 > v_2$

(or)

$h_2 < h_1$, $v_2 > v_1$

⇒ from this I can only say that police will overtake/catch theif, but I dont know about equal positions.



Let's assume both are at equal pos after some time 't'.

Initially at $t = 0$, h_1, h_2

after T's, $h_1' = h_1 + Tv_1$

$h_2' = h_2 + Tv_2$

We assumed that they meet after 'T' s

$\therefore \quad h_1' = h_2'$

$\Rightarrow h_1 + Tv_1 = h_2 + Tv_2$

$\Rightarrow Tv_1 - Tv_2 = h_2 - h_1$

$\Rightarrow T(v_1 - v_2) = h_2 - h_1$

$\Rightarrow T = (h_2 - h_1) / (v_1 - v_2)$

★ Can police catch theif ?

⇒ yes

★ But the Q is not about catching, we need to see at some time they are at same positions

$h_1 = 2m \qquad h_2 = 4m$

$v_1 = 7m/s \qquad v_2 = 2m/s$

$T = \frac{4-2}{7-2}$

$= \frac{2}{5} = 0.4s$ ✗

★ fractions are not possible

$T = 1s, 2s, 3s, 4s \ldots\ldots$

★ they will meet only when T is a natural number

(1s, 2s, 3s, 4s, ……)

④ $(h_2 - h_1) \,\%\, (v_1 - v_2) == 0$ ⇒ T / ↘ F

Q: 5m S: 7m

V_1: 3m/s V_2: 2m/s

t	h_1	h_2
t = 0	5m	7m
t = 1	8m, +3	9m, +2
t = 2	11m, +3	11m, +2
t = 3	14m, +3	13m, +2
t = 4	17m,	15m, +2
t = 10	35m,	—

If we need t = 4,

Initial h_1 = 5m

we need to do $+V_1$ 4 times

$\Rightarrow$ +3 4 times.

5m (+3) 4 times

$\Rightarrow$ 5 + 4*3

$\Rightarrow$ 5 + 12

= 17m

generalise,

$h_1' = h_1 + tV_1$

$h_2' = h_2 + tV_2$

★ How to Approach ?

1. understad Qs clearly
   - → Examples (ip, op)
   - → Constraints (Test cases are limited to this)

2. → put your thoughts on paper
   - → take multiple examples
   - → analyse and make observations
   - → Just think how a normal person solves.

3. write some steps / have some logic in mind
   - → test that with examples
     (try to take examples where you feel your approach/logic fails)

4. Code → errors / failed tests