

\* Check prime : (which is divisible only by 1 and itself)

eg:  $n = 5$   
op: true

eg:  $n = 13$   
op: true

Q: Check 13 is prime or not?

$\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, \cancel{5}, \cancel{6}, \cancel{7}, \cancel{8}, \cancel{9}, \cancel{10}, \cancel{11}, \cancel{12}, \cancel{13}$

[ $q - 12$ ] Is there any number  
that divides 13?

Q: check 49 is prime or not?

[ $q - 48$ ]

$$49 \div 2 = 0 \times$$

$$49 \div 3 = 0 \times$$

$$49 \div 4 = 0 \times$$

$$49 \div 5 = 0 \times$$

$$49 \div 6 = 0 \times$$

$$49 \div 7 = 0 \checkmark \quad (7 \times 7 = 49)$$

I found a  
number that  
divides 49  
 $\Rightarrow$  not prime

$\Rightarrow$  I need not check 8 - 48 (ignored)

$$13 \div 2 = 0 \times$$

$$13 \div 3 = 0 \times$$

$$13 \div 4 = 0 \times$$

$$13 \div 5 = 0 \times$$

$$13 \div 6 = 0 \times$$

$$13 \div 7 = 0 \times$$

$$13 \div 8 = 0 \times$$

$$13 \div 9 = 0 \times$$

$$13 \div 10 = 0 \times$$

$$13 \div 11 = 0 \times$$

$$13 \div 12 = 0 \times$$

$\Rightarrow$  I have tried and failed  
to find a number that  
divides 13  $\Rightarrow$  prime number

→ Convert to code,

I did not find any number  
that divides N.

```
let isFound = false;      num < N  
                        (or)  
for (let num = 2; num <= N-1; num++) {  
    if (N % num == 0) {  
        isFound = true;  
        break;  
    }  
}
```

```
}  
if (isFound == true) → console.log ("Not prime");  
else → console.log ("prime");
```

\* Check all

$N \% \text{num} == 0$

$\text{num} \rightarrow 2 \text{ to } N-1$

then only decide  
prime or not;

\* You cannot take  
prime or not  
decision inside  
the for loop.

⇒ flag variable Technique \*

## \* while Loop :

```
483 let num = 1;
484 while (num <= 5) {
485     console.log(`Level ${num}`);
486     num++;
487 }
```

- ⇒ initialisation, operation are handled separately.
- ⇒ you can use any loop, generally we will pick whichever is easy for the problem.

1. num = 1
2. num  $\leftarrow 5 \Rightarrow 1 \leftarrow 5 \Rightarrow \text{true}$   
→ "Level 1"  
→ num++  $\Rightarrow$  num = 2
3. num  $\leftarrow 5 \Rightarrow 2 \leftarrow 5 \Rightarrow \text{true}$   
→ "Level 2"  
→ num++  $\Rightarrow$  num = 3
4. num  $\leftarrow 3 \Rightarrow 3 \leftarrow 5 \Rightarrow \text{true}$   
→ "Level 3"  
→ num++  $\Rightarrow$  num = 4
5. num  $\leftarrow 4 \Rightarrow 4 \leftarrow 5 \Rightarrow \text{true}$   
→ "Level 4"  
→ num++  $\Rightarrow$  num = 5
6. num  $\leftarrow 5 \Rightarrow \text{true}$   
→ "Level 5"  
→ num++  $\Rightarrow$  num = 6
7. num  $\leftarrow 5 \Rightarrow \text{false}$

\* A problem which can be solved using for loop can also be solved using while loop and vice versa.

\* Sum of digits :

Ex: 1325

Op:  $1+3+2+5 = 11$

Ex: 565423

Op:  $5+6+5+4+2+3 = 25$

\* extract the last digit,

$$\text{lastDigit} = \text{num} \% 10;$$

\* remove last digit,

$$\text{num} = \text{num}/10$$

$\Rightarrow$  we repeat this process until num becomes 0.

$$\text{num} = \begin{array}{cccc} 1 & 3 & 2 & 5 \\ a & b & c & d \end{array}$$

$a=1$   
 $b=3$   
 $c=2$   
 $d=5$

$$\Rightarrow a+b+c+d$$

$$1325 \% 10 = \boxed{5}$$

$$\downarrow 132 \cancel{*} \quad 1325/10 = 132.5 \\ = 132$$

$$132 \% 10 = \boxed{2}$$

$$\downarrow 13 \cancel{*} \quad 132/10 = 13.2 \quad \text{Panee Int} \\ = 13$$

$$13 \% 10 = \boxed{3}$$

$$\downarrow 1 \cancel{*} \quad 13/10 = 1.3 = 1$$

$$1 \% 10 = \boxed{1}$$

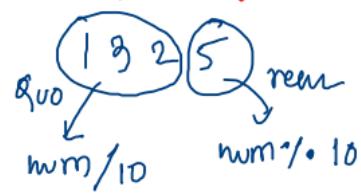
$$\downarrow * \quad 1/10 = 0.1 \\ = 0$$

0 (stop)

\* how to extract each digit from a number

$$\begin{array}{r} +5 \\ +2 \\ +3 \\ +1 \\ \hline = \boxed{11} \end{array}$$

\* Digit extraction Technique



\* Reverse of a Number :

Eg: 1325

Op: 5231

$$\begin{array}{cccc}
 1000 & 100 & 10 & 1 \\
 | & 8 & 2 & 5 \\
 \diagdown & \diagup & \diagdown & \diagup \\
 1 & 3 & 2 & 5 \\
 \diagup & \diagdown & \diagup & \diagdown \\
 1000 & 100 & 10 & 1
 \end{array}$$

$\Rightarrow 1 \times 1000 + 3 \times 100 + 2 \times 10 + 5 \times 1$

$$\begin{aligned}
 &= 5 \times 1000 \\
 &\quad + 2 \times 100 \\
 &\quad + 3 \times 10 \\
 &\quad + 1 \times 1
 \end{aligned}$$

\*  $\text{rev} = \text{rev} \times 10 + \text{digit}$

$$1325 \mod 10 = 5$$

$$\downarrow 1325/10$$

$$132 \mod 10 = 2$$

$$\downarrow 132/10$$

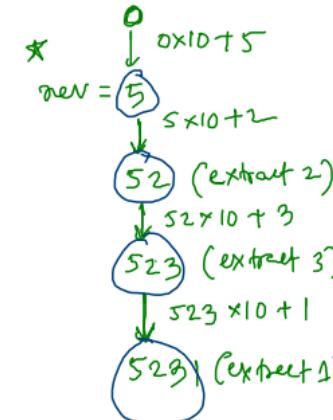
$$13 \mod 10 = 3$$

$$\downarrow 13/10$$

$$1 \mod 10 = 1$$

$$\downarrow 1/10$$

$$0$$



whenever 2 is extracted, when 3 is extracted

$$\text{rev} = 5 \rightarrow 52$$

$$\begin{aligned}
 &5 \times 10 + 2 \\
 &\Rightarrow 50 + 2 = 52
 \end{aligned}$$

$$\text{rev} = 52 \rightarrow 523$$

$$\begin{aligned}
 &52 \times 10 + 3 \\
 &\Rightarrow 520 + 3 = 523
 \end{aligned}$$

Whenever 1 is extracted

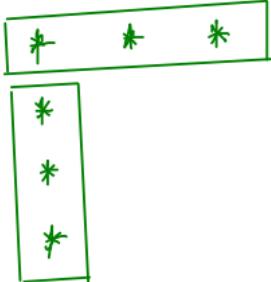
$$523 \rightarrow 5231$$

$$\begin{aligned}
 &523 \times 10 + 1 \Rightarrow 5230 + 1 \\
 &\Rightarrow 5231
 \end{aligned}$$

## \* N stars :

Eg :  $n = 3$

Op:



```
for(let i=0; i<n; i++) {
    process.stdout.write(" *");
}
```

$n = 3$

```
}
console.log(); // dummy
```

```
for(let i=0; i<n; i++) {
    console.log(" *");
}
```

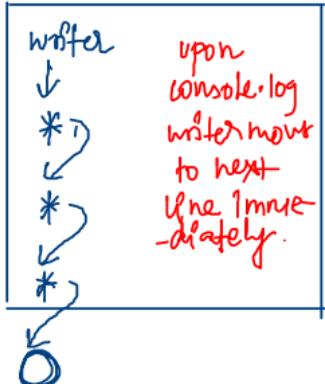
$i = 0$   
 $psw(" *")$

$i = 1$   
 $psw(" *")$

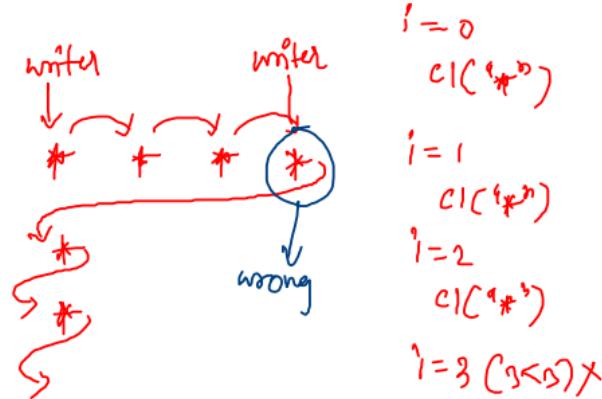
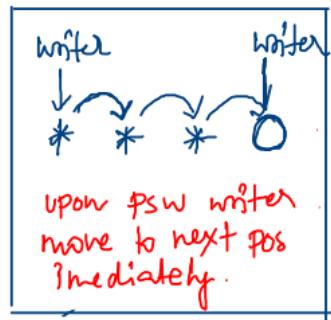
$i = 2$   
 $psw(" *")$

$i = 3 (3 < 3) X$

Notes / console / op



Notes / console / op



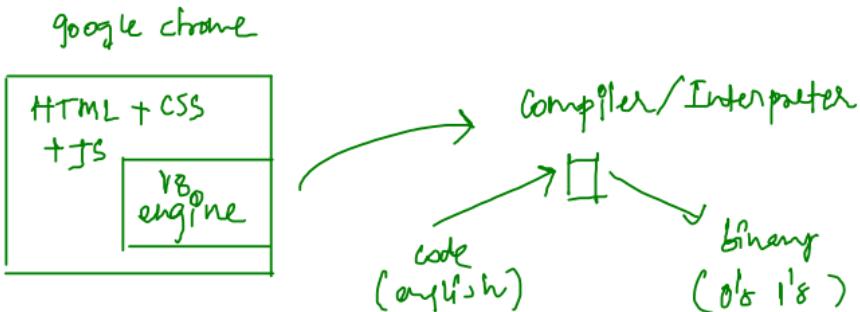
$i = 0$   
 $c1(" *")$

$i = 1$   
 $c1(" *")$

$i = 2$   
 $c1(" *")$

$i = 3 (3 < 3) X$

\* Process · stdout · write :



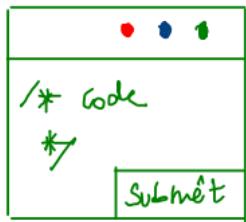
c/c++ cmd	java JVM	python python
mingw		

⇒ Installing a software  
before running  
programs

J8  
→ running it on  
google chrome  
→ google is acting as  
a compiler.

(frontend)  
(Your computer)

online Ide



(USA)

API call

Js Code Request

output Response

→ this happens through  
multiple connections over  
Internet

\* process.stdout.write and  
readline are part of libuv.

→ As chrome doesn't have libuv it  
cannot understand about process.

( Some computer over  
Internet )

Backend Server

the code you  
sent will  
run here and  
ip is sent back

V8 engine  
+ libuv

(India)

how will backend  
server run JS code?

→ Node.js provides the  
V8 engine

→ Node.js is a runtime  
environment for JS.  
(a playground for JS)