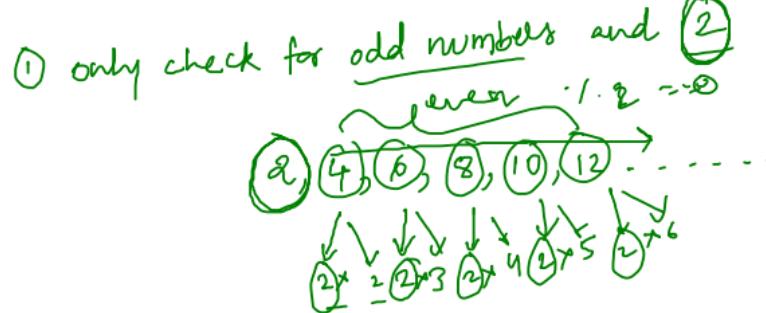


## # Improving check prime :



$$n = 37$$

\* approx  $\frac{n}{2}$   
iterations  
 $\Rightarrow \frac{n}{2}$  checks

$$37 \cdot 1 \cdot 2 = 0 \rightarrow 2$$

$$37 \cdot 1 \cdot 3 = 0$$

$$37 \cdot 1 \cdot 5 = 0$$

$$37 \cdot 1 \cdot 7 = 0$$

⋮

$$37 \cdot 1 \cdot 35 = 0$$

$\left. \begin{matrix} 37 \cdot 1 \cdot 3 \\ 37 \cdot 1 \cdot 5 \\ 37 \cdot 1 \cdot 7 \\ \vdots \\ 37 \cdot 1 \cdot 35 \end{matrix} \right\} \text{odd numbers}$

converting all the even numbers

~~④  $n \% 2 == 0$~~

~~$n \% 4 == 0$~~

~~$n \% 6 == 0$~~

~~$n \% 8 == 0$~~

~~$n \% 10 == 0$~~

~~$n \% 12 == 0$~~

~~$n \% 14 == 0$~~

~~$n \% 16 == 0$~~

~~$n \% 18 == 0$~~

~~$n \% 20 == 0$~~

~~$n \% 22 == 0$~~

~~$n \% 24 == 0$~~

~~$n \% 26 == 0$~~

~~$n \% 28 == 0$~~

~~$n \% 30 == 0$~~

~~$n \% 32 == 0$~~

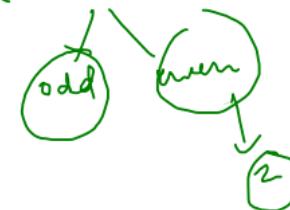
~~$n \% 34 == 0$~~

~~$n \% 36 == 0$~~

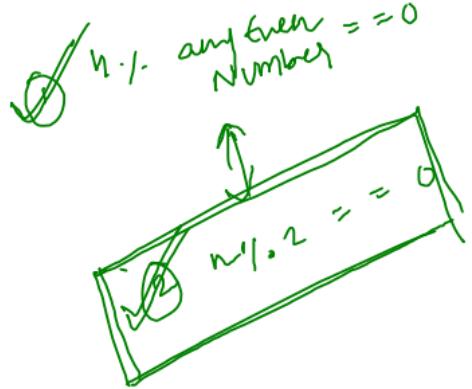
~~$n \% \text{anyEvenNumber} == 0$~~

ignore

$$37 (2 \rightarrow 36)$$



36



$$36 \cdot 1 \cdot 2 = = 0 \quad (2 \times 18)$$

even number

~~$$36 \cdot 1 \cdot 4 = = 0 \quad (4 \times 9)$$~~

~~$$36 \cdot 1 \cdot 6 = = 0 \quad (6 \times 6)$$~~

~~$$36 \cdot 1 \cdot 8 = = 0 \quad \times$$~~

~~$$36 \cdot 1 \cdot 10 = = 0 \quad \times$$~~

~~$$36 \cdot 1 \cdot 12 = = 0 \quad (12 \times 3)$$~~

~~$$36 \cdot 1 \cdot 14 = = 0 \quad (2 \times 18)$$~~

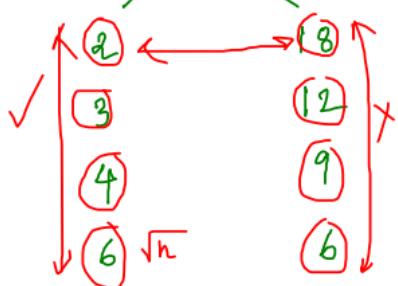
$$\textcircled{1} \textcircled{n} \cdot \textcircled{14} = = 0$$

$$\textcircled{n} \cdot \textcircled{2} = = 0$$

↑  
even

②

$$n = 36$$



$$36 \Rightarrow [2, 35] \\ \Rightarrow [2, 6]$$

$$[2, \sqrt{n}]$$

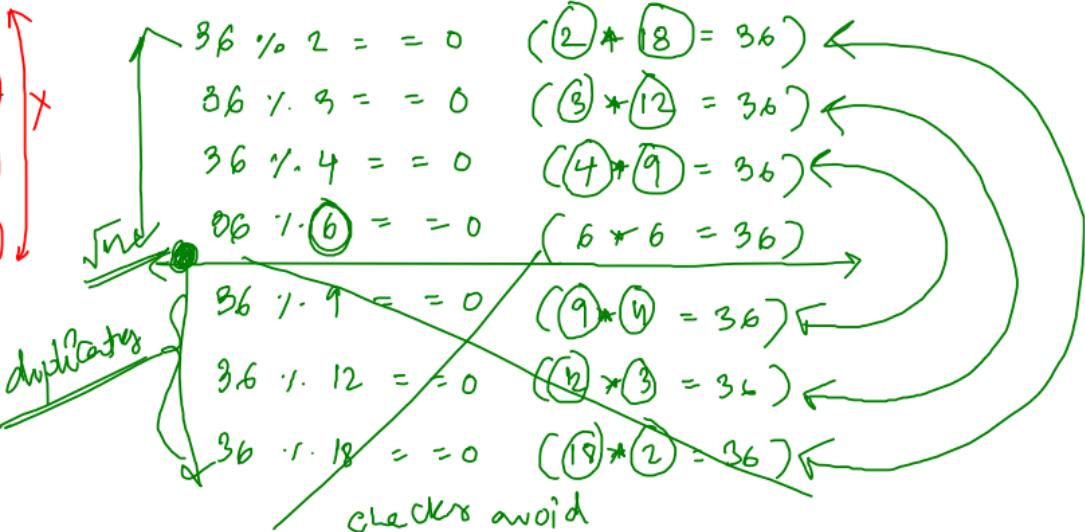
$$[2, \sqrt{36}] \Rightarrow [2, 6]$$

$$36 \cdot 4 = 0$$

↑  
Same only

$$36 \cdot 9 = 0$$

↓  
 $9 \cdot 4 = 36$



$\Rightarrow$  after  $\sqrt{n}$  all the checks are duplicates of your previous checks

$$\eta = 24$$

$$\begin{array}{c} 2 \\ 3 \\ 4 \end{array} \quad \begin{array}{c} 12 \\ 8 \\ 6 \end{array}$$

$$\begin{array}{l} \textcircled{(2, 4)} \\ \textcircled{(2, 23)} \end{array}$$

$$[2, \sqrt{24}]$$

$$[2, 4]$$

$$\begin{array}{ll} 24 \cdot 1 \cdot 2 = -0 & (2 * 12 = 24) \\ 24 \cdot 1 \cdot 3 = -0 & (3 * 8 = 24) \\ 24 \cdot 1 \cdot 4 = -0 & (4 * 6 = 24) \\ \cancel{24 \cdot 1 \cdot 6 = -0} & (\cancel{6 * 4 = 24}) \\ \cancel{24 \cdot 1 \cdot 8 = -0} & (\cancel{8 * 3 = 24}) \\ \cancel{24 \cdot 1 \cdot 12 = -0} & (\cancel{12 * 2 = 24}) \end{array}$$

same  
duplicator  
avoid false checker

→  $[2, \sqrt{n}]$

no. of checks / iterations =  $\sqrt{n}$  approx

→ we are performing break

if  $n \cdot i \cdot x == 0$

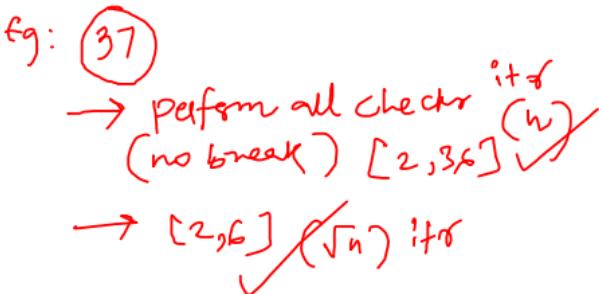
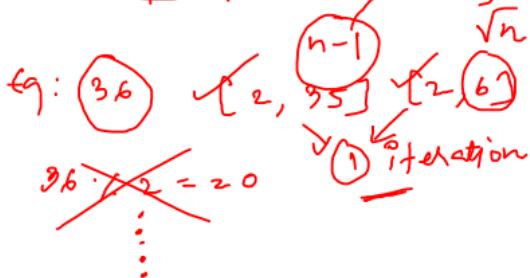
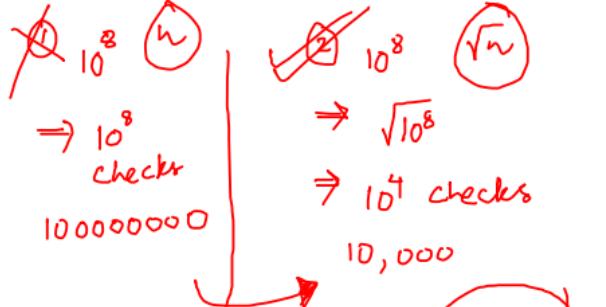
for(  $x = 2$  ;  $x \leq \boxed{n-1}$   $x++$ )

    if ( $n \cdot i \cdot x == 0$ ) {

        break;

}

→ so this difference of no. of iterations  
is much evident when number is prime



Q: Sum of digits

Eg: 1 3 2 5

Op:  $(1+3+2+5) = \underline{\underline{11}}$

Eg: 5 6 5 4 2 3

Op: 25  $(5+6+5+4+2+3)$

$$132\cancel{5} \% \cdot 10 = 5$$

$$\downarrow 1325 \% \cdot 10 = 132$$

$$13\cancel{2}\cancel{5} \% \cdot 10 = 2$$

$$\downarrow 132 \% \cdot 10 = 13$$

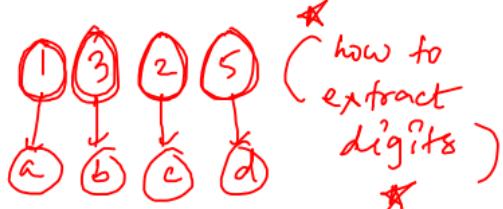
$$1\cancel{3}\cancel{2}\cancel{5} \% \cdot 10 = 3$$

$$\downarrow 13 \% \cdot 10 = 1$$

$$1\cancel{3} \% \cdot 10 = 1$$

$$\downarrow 1 \% \cdot 10 = 0$$

$$0 \times$$



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

$n \% \cdot 10 \rightarrow$  last digit ~~remove~~  
 $\rightarrow$  removing last digit  $\rightarrow (n/10)$

$$\text{Sum} = \varphi$$

$$\begin{array}{c} \nearrow \\ x \\ \times \\ 10 \\ 11 \end{array}$$

```

403 function SumofDigits(n) {
404     let sum = 0;
405     while (n != 0) {
406         const digit = n % 10; // get last digit
407         sum = sum + digit;
408         n = parseInt(n / 10); // remove last digit
409     }
410
411     console.log(sum);
412 }
```

\* how many iterations  
 $= \log_{10} n + 1$  (optional)

$$n = 1234 \text{ (4 iterations)}$$

$$n = 12 \text{ (2 iterations)}$$

$$n = 1 \ 2 \ 3 \ 4$$

$$\begin{array}{l} \text{sum} = \cancel{\emptyset} \\ \cancel{4} \\ \cancel{1} \\ \cancel{9} \\ \textcircled{10} \end{array}$$

$$\textcircled{1} \quad 1234 \underset{!}{=} 0$$

$$1234 \cancel{.} \cdot 10 = 4$$

$$n = 1234 \cancel{/10} = 123$$

$$\textcircled{2} \quad 123 \underset{!}{=} 0$$

$$123 \cancel{.} \cdot 10 = 3$$

$$n = 123 \cancel{/10} = 12$$

$$\textcircled{4} \quad 1 \underset{!}{=} 0$$

$$1 \cancel{.} \cdot 10 = 1$$

$$\textcircled{3} \quad 1 \underset{!}{=} 0$$

$$1 \cancel{.} \cdot 10 = 0$$

$$n = \cancel{1} \cancel{/10} = 0$$

$$n = 12 \cancel{/10} = 1$$

$$\textcircled{5} \quad 0 \underset{!}{=} 0 \times$$

Q: Reverse of a number :

eg: 1 3 2 5

op: 5 2 3 1

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

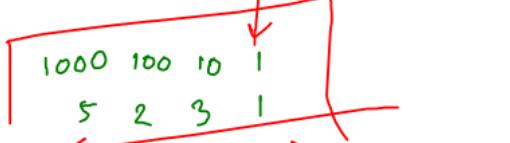
$$\Rightarrow 5 * 1 + 2 * 10 + 3 * 100 + 1 * 1000$$

$$\Rightarrow 5 + 20 + 300 + 1000$$

$$\Rightarrow 1325$$

$$\begin{aligned} 1325 \div 10 &= 5 * 1000 \\ 132 \div 10 &= 2 * 100 \\ 13 \div 10 &= 3 * 10 \\ 1 \div 10 &= 1 * 1 \end{aligned}$$

$\times 0$



$$\Rightarrow \underline{1} * 1 + 3 * 10 + 2 * 100$$

$$\Rightarrow 1 + 30 + 200 + 5000$$

$$\Rightarrow \underline{\underline{5231}}$$

$$\Rightarrow \underline{\underline{5 * 1000}}$$

rev = 0

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

$$\textcircled{1} \quad 1325 \div 10 = 5$$

$$n = 1325 / 10 = 132$$

$$\text{rev} = 0 * 10 + 5 = 5$$

$$\textcircled{2} \quad 132 \div 10 = 2$$

$$n = 132 / 10 = 13$$

$$\text{rev} = 5 * 10 + 2 = 52$$

$$\textcircled{3} \quad 13 \div 10 = 3$$

$$n = 13 / 10 = 1$$

$$\text{rev} = 52 * 10 + 3 = 523$$

$$\textcircled{4} \quad 1 \div 10 = 1$$

$$n = 1 / 10 = 0$$

$$\text{rev} = 523 * 10 + 1 = 5231$$

$$\textcircled{5} \quad \times n_{\frac{1}{2}} = 0$$

```

421 readline.question("", (n) => {
422   let rev = 0;
423   while (n != 0) {
424     const digit = n % 10;
425     n = parseInt(n / 10);
426     rev = rev * 10 + digit;
427   }
428   console.log(rev);
429   readline.close();
430 });

```

$$n = 1325, \text{ rev} = \emptyset \neq 523 \quad (523)$$

$$\textcircled{1} \quad 1325 \div 10 = 0$$

$$\begin{aligned} \text{digit} &= 1325 \div 10 \\ &= 5 \end{aligned}$$

$$n = 1325 \div 10 = 132$$

$$\text{rev} = 0 * 10 + 5 = 5$$

$$\textcircled{3} \quad 13 \div 10 = 0$$

$$d = 13 \div 10 = 3$$

$$n = 13 \div 10 = 1$$

$$\text{rev} = 52 * 10 + 3 = 523$$

$$\textcircled{2} \quad 132 \div 10 = 0$$

$$d = 132 \div 10 = 2$$

$$n = 132 \div 10 = 13$$

$$\text{rev} = 5 * 10 + 2 = 52$$

$$\textcircled{4} \quad 1 \div 10 = 0$$

$$d = 1 \div 10 = 1$$

$$n = 1 \div 10 = 0$$

$$\text{rev} = 523 * 10 + 1 = 5231$$

$$\textcircled{5} \quad 0 \div 10 \neq 0$$

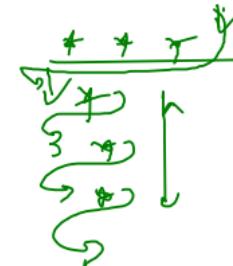
Q: N stars:

Eg:  $n = 3$

\* \* \* → horizontal

\* \* \*  
} vertical

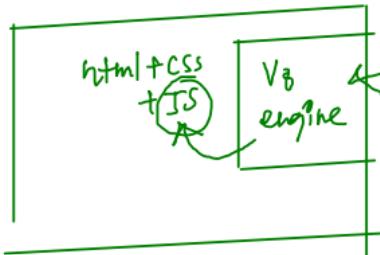
```
439 | readline.question("", (n) => {  
440 |   // Write your code here  
441 |  
442 |   // horizontal stars  
443 |   for (let i = 0; i < n; i++) {  
444 |     process.stdout.write("*");  
445 |   }  
446 |  
447 |   console.log();  
448 |  
449 |   // vertical stars  
450 |   for (let i = 0; i < n; i++) {  
451 |     console.log("*");  
452 |   }  
453 |   readline.close();  
454 |});
```



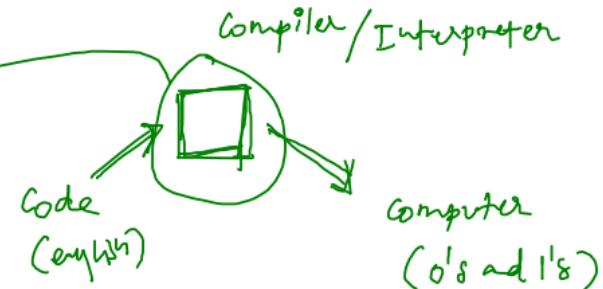
\* process.stdout.write

⇒ This is not related to JS, this is related to node.js.

google chrome



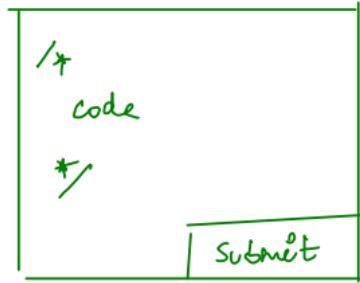
mingw JDK python  
c/c++/Java/python



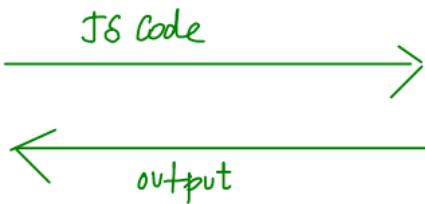
① Install compiler

① web browser

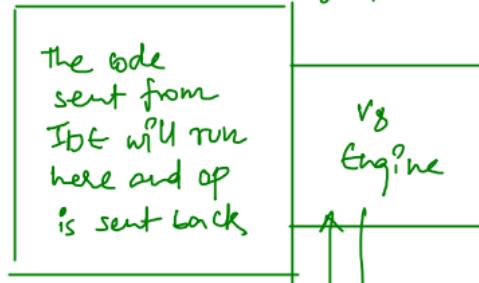
online IDE



friend(1)



Backend  
server ( some other  
computer  
over internet )



friend(2)

how will you get  
V8 engine to run  
JS on server

Node.js

( runtime environment  
for JS )