Pattern

n = 7

```
1 2 3 4 5 6 7
 2 3 4 5 6 7
  3 4 5 6 7
   4 5 6 7
    5 6 7
     6 7
      7
     6 7
    5 6 7
   4 5 6 7
  3 4 5 6 7
 2 3 4 5 6 7
1 2 3 4 5 6 7

Input: n = 7
```

line 4 → ___4_5_6_7

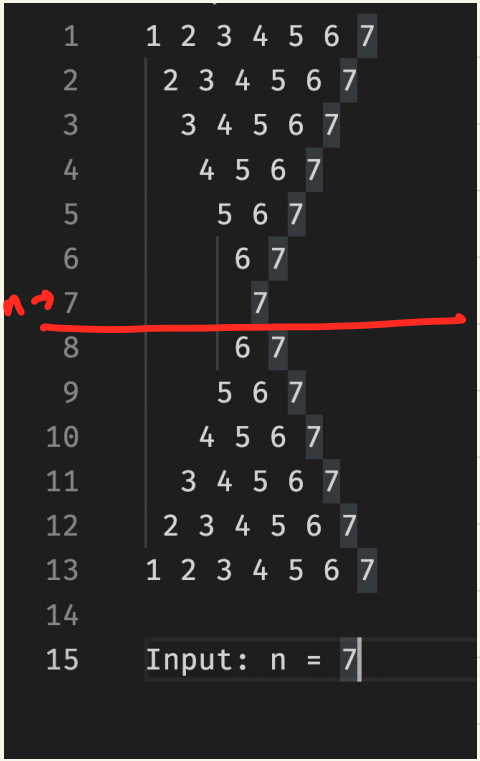
upper

lower

line x → _____ x . x+1 _ x+2 ... n

x-1 spaces

line	spaces
1	0
2	1
3	2
4	3
⋮	
x	x-1



Repeat a process for every row

↳ for any row x.

① print x-1 spaces

② After it, start printing numbers from x to n space separated

if ≤

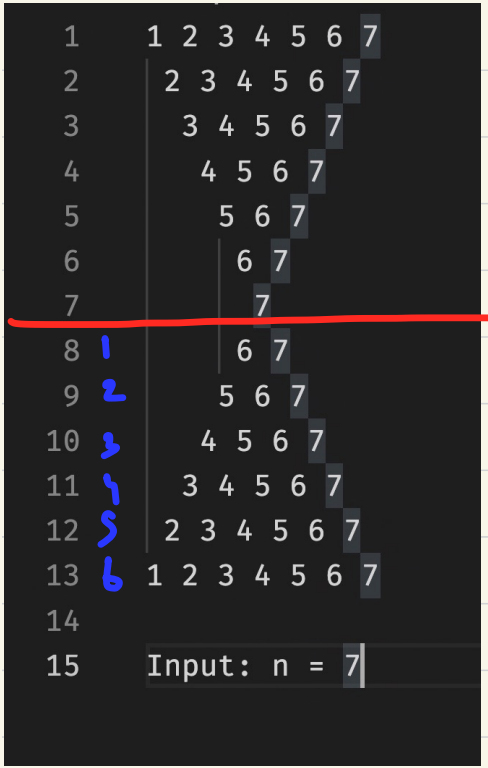
n=7

```
for (let row=1 ; row <= n ; row++) {
    let spaces = row-1;
    let str = "";
    for (let i=1; i<= spaces ; i++) {
        str += " ";
    }
    let num = row;
    while (num <= n) {
        str += num + "__";
        num++;
    }
    console.log (str);
}
```

str = "--3_4_5_6_7_"

row=3

num=3 4 5 6 7



lower

Row	Spaces
1	5
2	4
3	3
4	2
5	1
6	0

(n-row-1)

Every row starts with → (n-row) and you tell n, Spaces Separated

In the lower part how many rows we have to Print ??

n = 7 → 6 rows → n-1 rows

```
for (let row=1 ; row<=n-1 ; row++){
    let str="";
    let spaces = n-row-1;
    for(let i=1; i<= spaces; i++){
        str += " ";
    }
    let num = n-row ;
    while (num <= n){
        str += num + " ";
        num++
    }console.log(str);
}
```

finalans: ~~0~~ ~~1~~ ~~2~~ 4

[1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0]

i

currentones: ~~0~~ ~~1~~ 2

if before hitting a zero we were counting a set of ones then they will break now.

Ex → [0, 0, 1, 0] ⟸

1

Ex → [0, 0, 0]

↳ 0

if we hit a one, ① we are already some consecutive ones or we were already lonely consec ones & we got one more

[1, 1, 0, 1, 1, 1]

i

finalAns = ~~0~~ 2

consecutive: ~~0 1 2~~

3

Every value present in the array is going to be in the range [1, n]

Return an array containg all the missing no. from the range [1, n].

[4, 3, 2, 7, 8, 2, 3, 1]

→ { key : value
4 : 4
3 : 3
2 : 2
7 : 7
8 : 8
1 : 1 }

for (int i=1 ; i<=n ; i++)

i = X X X X X X X

unique value

[5, 6]

num[i] → [1,n]

[-4, -3, -2, -7, 8, 2, -3, -1]

index → 0 1 2 3 4 5 6 7

Curr value = x → |x| Math.abs(x)

goto → index (x-1) & make it a negative value.

if we detect positive

|x| → x

(-x) → x

How to negative ?:

if (arr[x-1] > 0)

arr[x-1] *= -1;

res.push(i+1);

There won't be any -ve no.

When we were able to use extra space we were able to record those values which are present in the array.

So without extra space how can you do it <?>