Q. Given an array of integers, and a number x. Write a function which is recursive in nature to return the last index on which x is present in the array. Return -1, if it is not present.

arr → [6, 3, 1, 2, 3, 9, 3, -1]

x = 3

ans → 6

Base Case → if the whole array has been checked & we didn't find x, return -1.

→ if (i == -1)

Self Work → if (arr[i] == x) ??

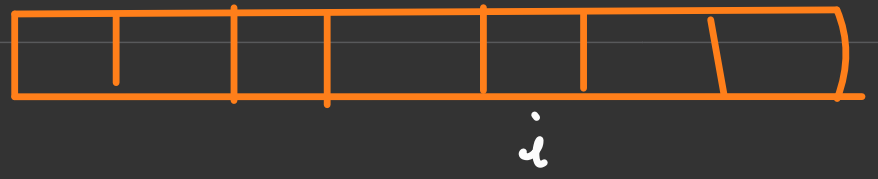
Assumption → assume → f(arr, x, i-1) works correctly

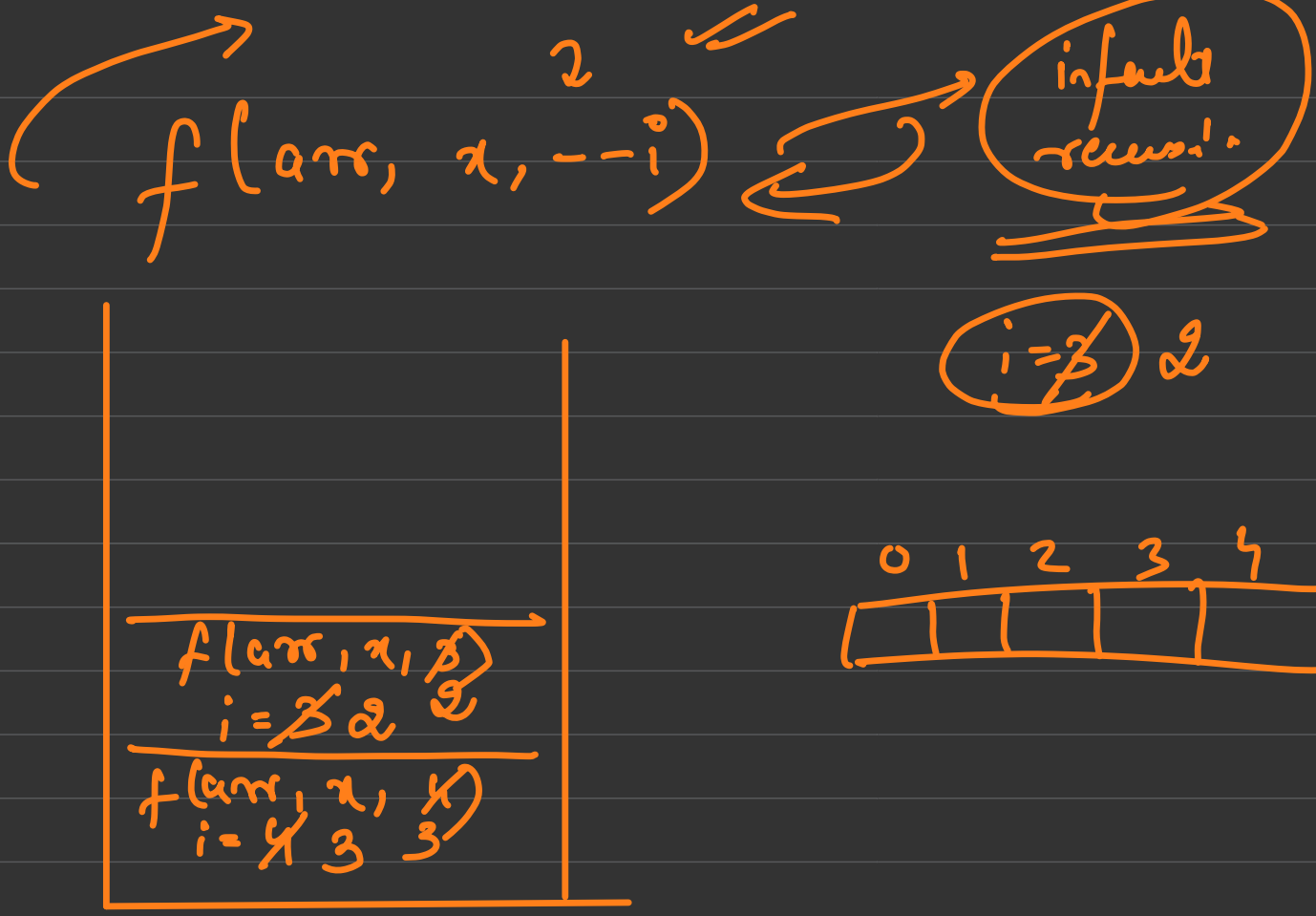
$f(arr, x, i)$ = $(arr[i] == x)$? i : $f(arr, x, i-1)$

the function f checks whether the element x is present in the array in the range $[0, i]$ or not

x is present at current index i.

ans - $f(arr, x, arr.length-1)$

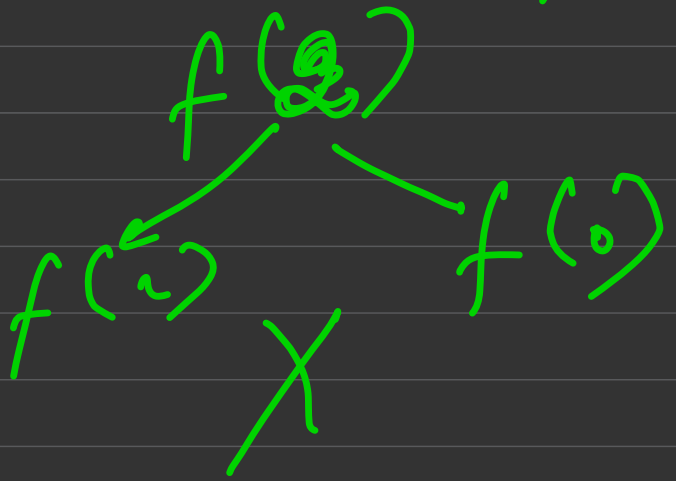



f(arr, x, --i)
2
infinite recursion
i = 3 2
f(arr, x, 4)
i = 4 3 3
0 1 2 3 4

$f(n) = f(n-1) + f(n-2)$

$f(3)$

$f(--n) + f(n-2)$

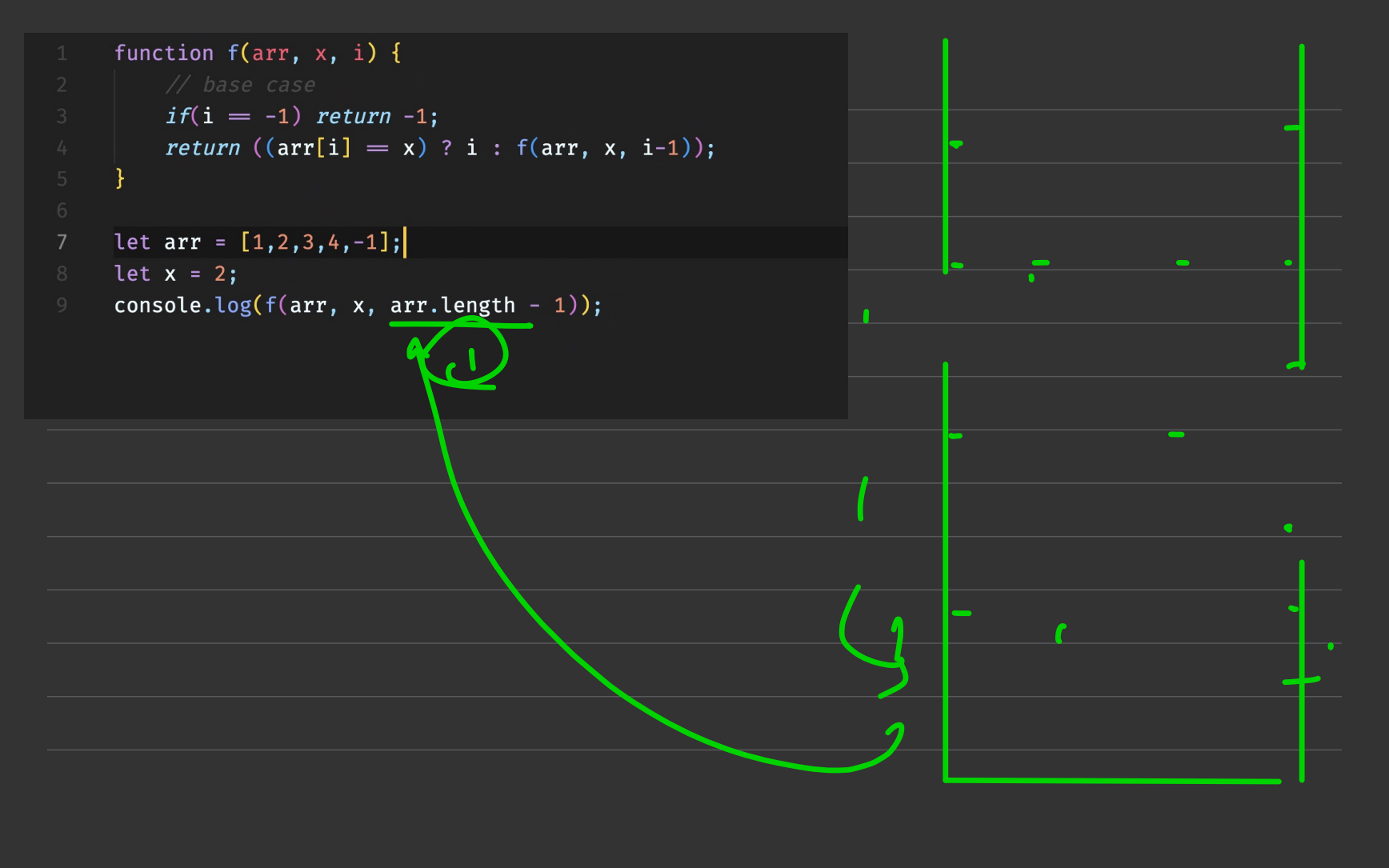


$$x = i - 1 \qquad i = 3$$

$$y = i - 1 \qquad i = 3$$

```
function f(arr, x, i) {
    // base case
    if(i == -1) return -1;
    return ((arr[i] == x) ? i : f(arr, x, i-1));
}

let arr = [1,2,3,4,-1];
let x = 2;
console.log(f(arr, x, arr.length - 1));
```



Q: Given a number x, find the sum of digits of x recursively.

Ex → x = 1234

ans → 10

x % 10 → last digit

↳ x/10 → remove the last digit

→ Sum of digits of 1234

ans will be → (sum of digits of 123) + 4

$f(x) = f(\text{floor}(x/10)) + x\%10$

the function returns the sum of digits of x.

sum of remaining digits

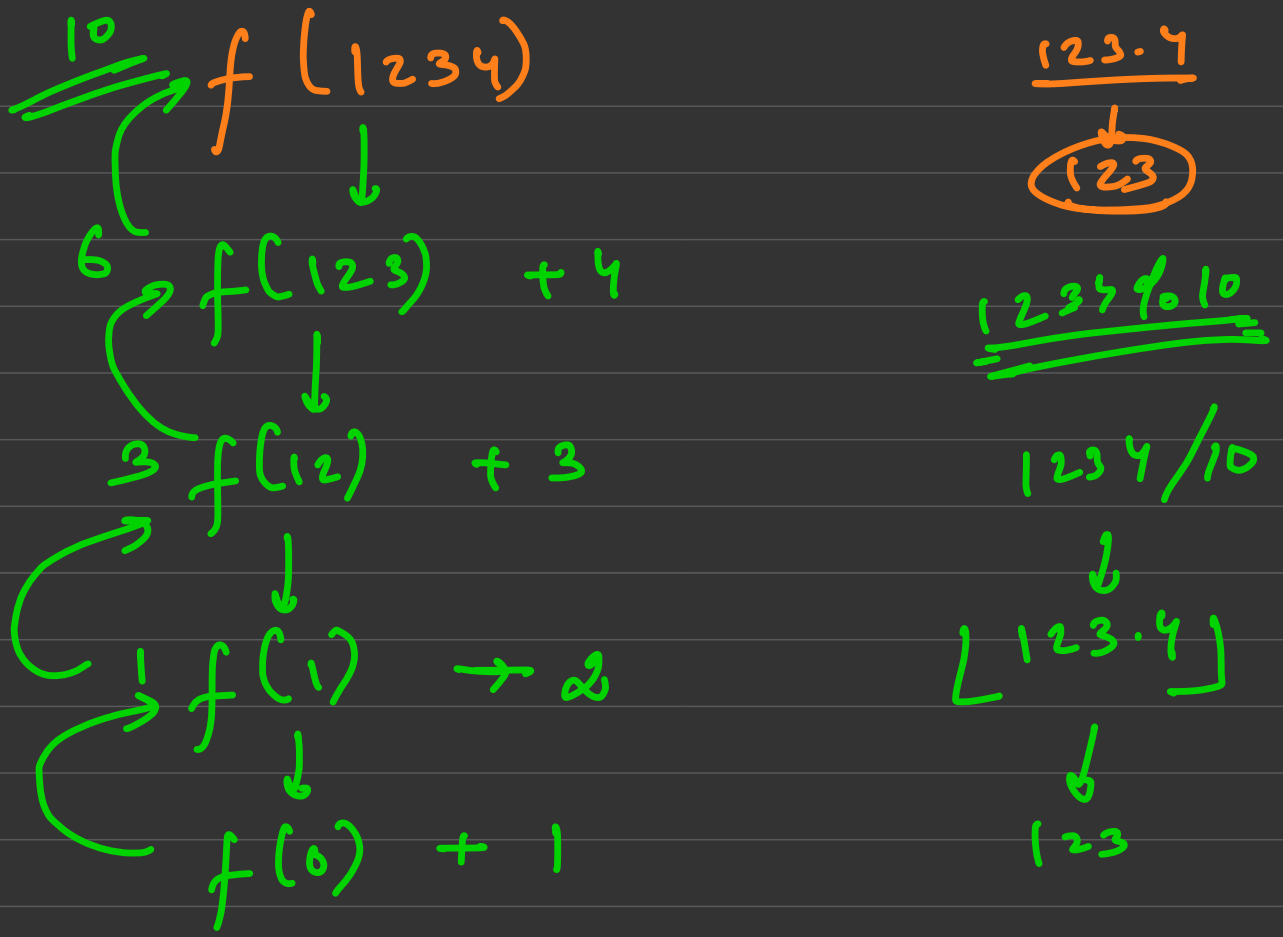
last digit

$1234 \rightarrow (123) + 4$

Base Case → if the no. is 0, Sum is 0:

assumption → f(floor(x/10)) works correctly

Self work → add the last digit to the sum of remaining digits:


10
f(1234)
6
f(123) + 4
3
f(12) + 3
1
f(1) → 2
f(0) + 1
123.4
123
1234%10
1234/10
⌊123.4⌋
123

Q. Print the following pattern, for given value of n, recursively.

n = 4

n = 5


```
for(i=1; i<=n; i++)
    for(j=1; j<=i; j++)
```

i → row

j → column

no. of stars per row == row number

Say we are printing row number 3,
when do we shift to row number 4??

↳ when last colum printed is col 3.

(col > row) → move to next row.

col → current col
row → current row

f(n, row, col) =>

↓

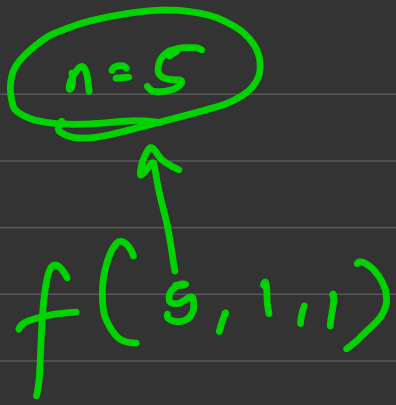
this function prints the pattern of triangle stars

n → total no. of rows expected

row → current row

col → current col for the current row.

```
if (row < col) {
    ans += "\n";
    f(n, row+1, 1)
} else {
    ans += "*"
    ↓
    f(n, row, col+1)
}
```



$f(n,1,1)$ → general ans

To move to the next row, we move to row+1, & restart column with 1. → $f(n, row+1, 1)$

→ To keep on printing the current row, we can just move to the next column → $f(n, row, col+1)$

Q. Print the following pattern, for given value of n, recursively.

→ n = 5

1 ★ ★ ★ ★ ★

2 ★ ★ ★ ★

3 ★ ★ ★

4 ★ ★

5 ★

no. of stars per row = = n - currow + 1

col > n - row + 1

Q. Write a recursive function that takes a string as an input and returns a new string having all the consecutive duplicates removed.

Ex → hello

ans → helo

→ hellooool

ans → helolol

i == i+1

donot consider i

else → consider i

hellolloolloll , output: hellolol

i

[str[i] == str[i+1]