RCA

Playlist

Loops

IS → 2 → on call DRI

apple music

Playlist consist of set of music files

for every music file we're showing a tile

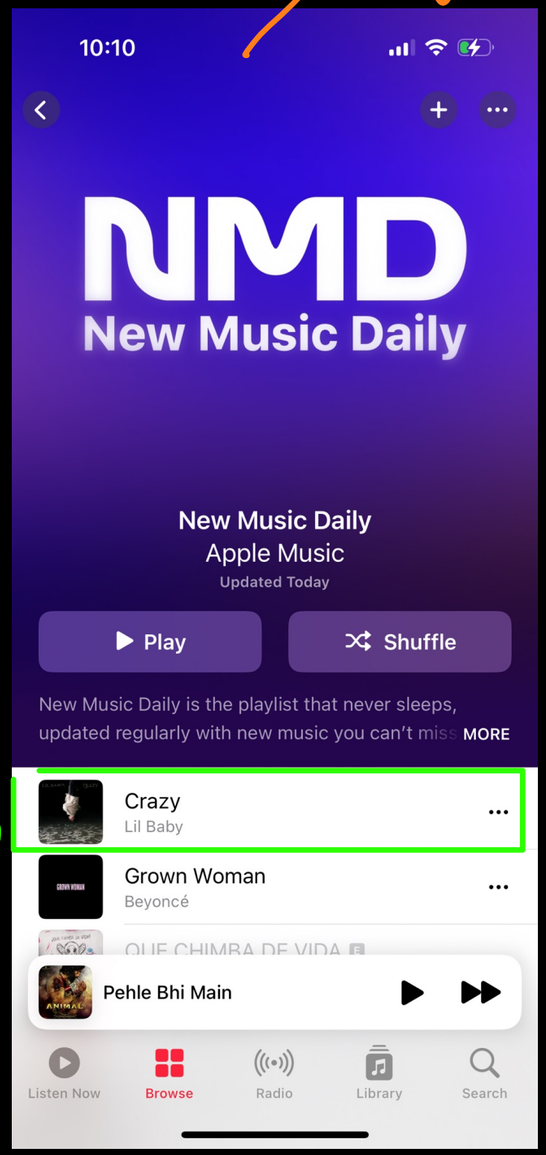
1000 songs

No

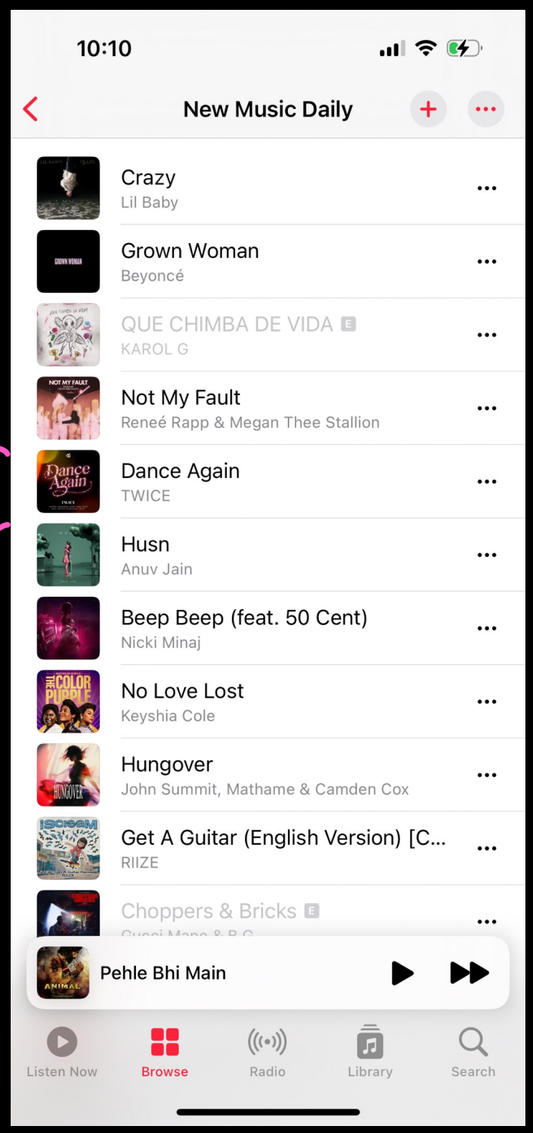
drawTile(...) manually 1000 times ?!

tile

drawTile(...)



not all in the user's cache

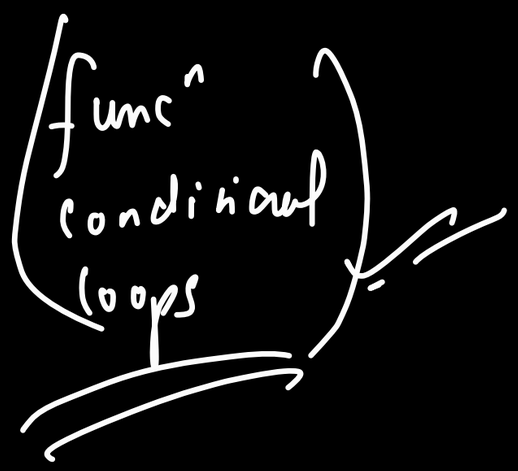


r3

Nature of repetition → For every music file we call drawTile with diff args.

↓

Can we automate the repetition

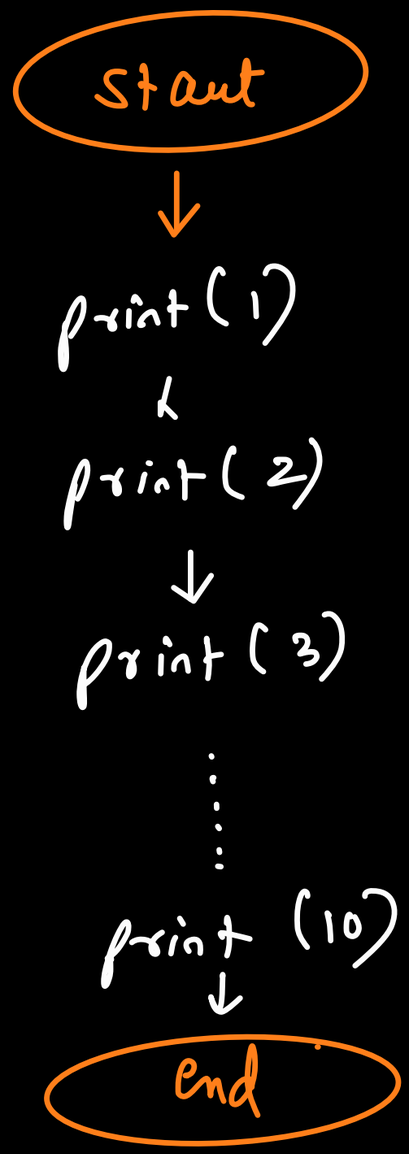


↓

Loops → loops are programming construct that helps you avoid manually doing repeated task.

flowchart:

print first 10 natural no.



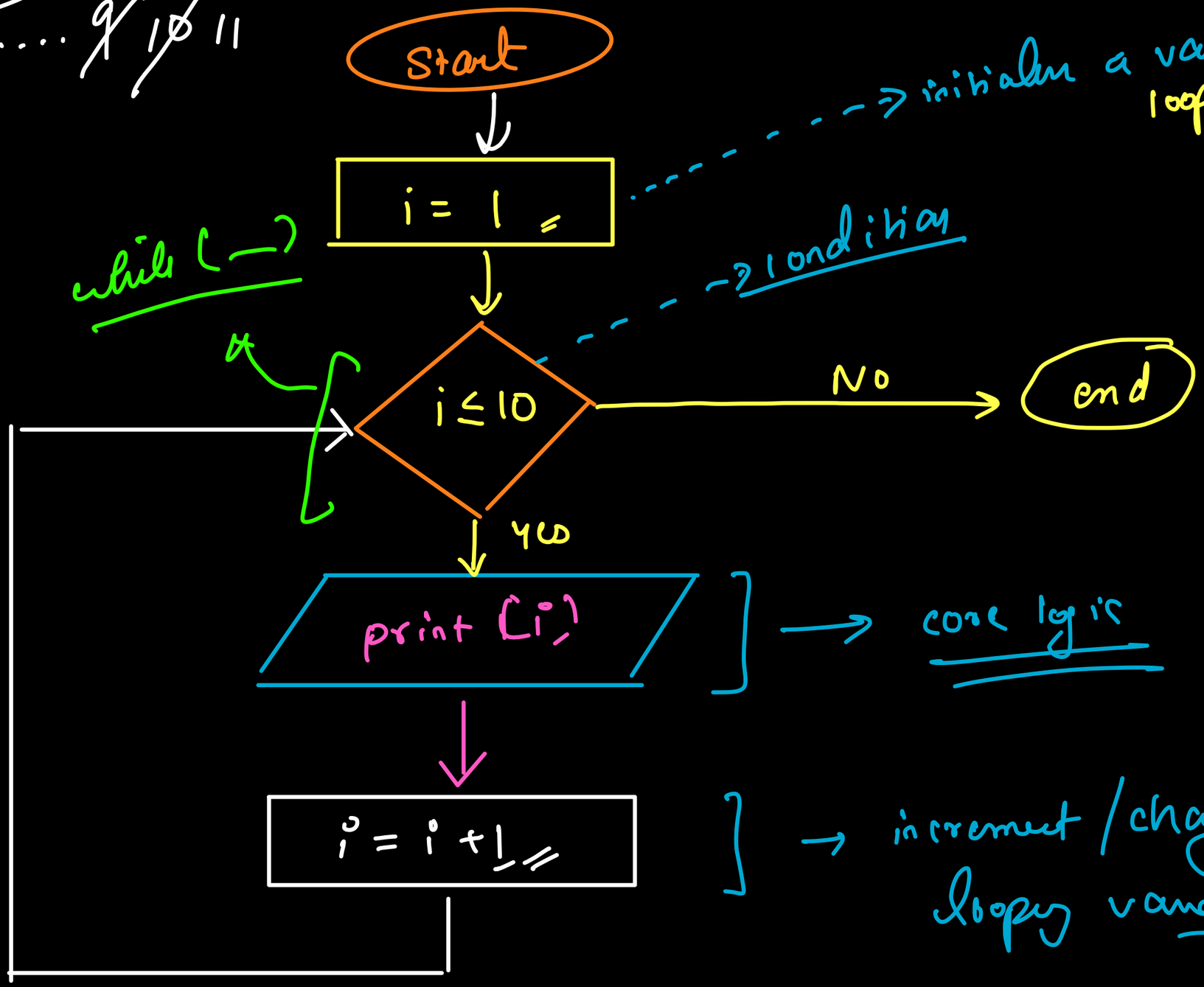
manual work

Sense of repetition is there

Calling print func^n

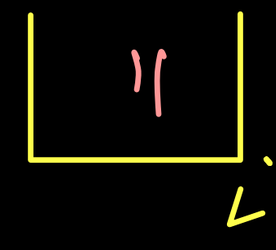
i = 1 2 3 4 5 .... 9 10 11

1
2
3
4
5
:
9
10



```
1)  let i = 1;
2)  while (i <= 10) {
3)      console.log(i);
4)      i = i + 1;
5)  }
```

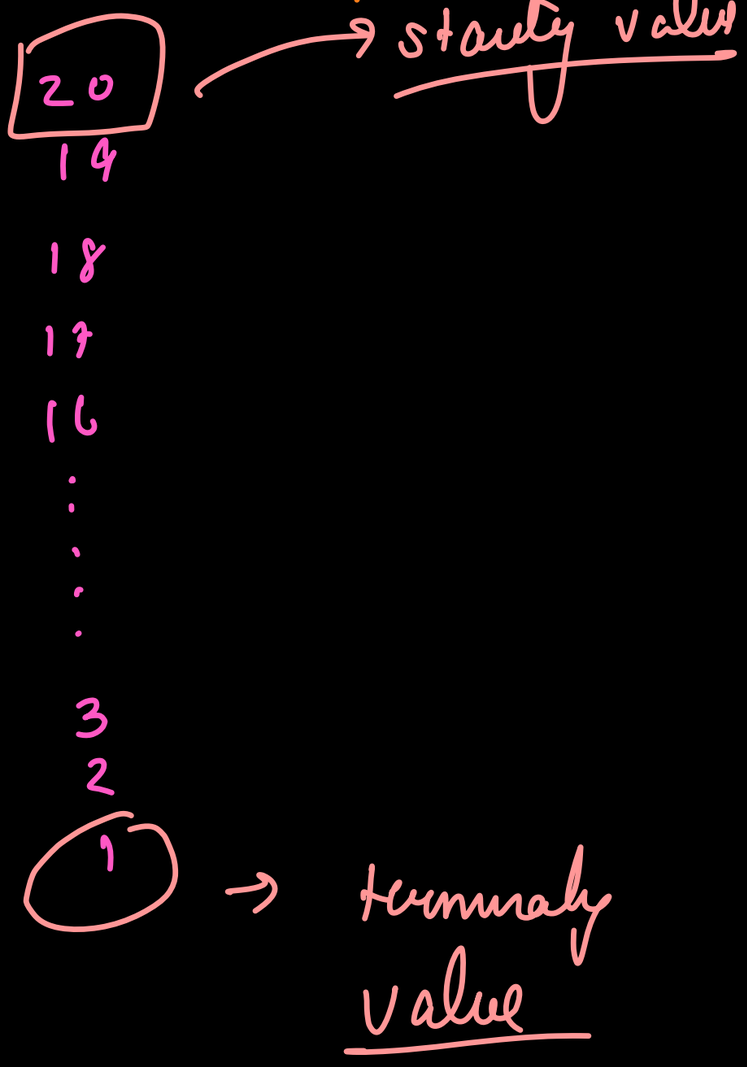
6) → ↵



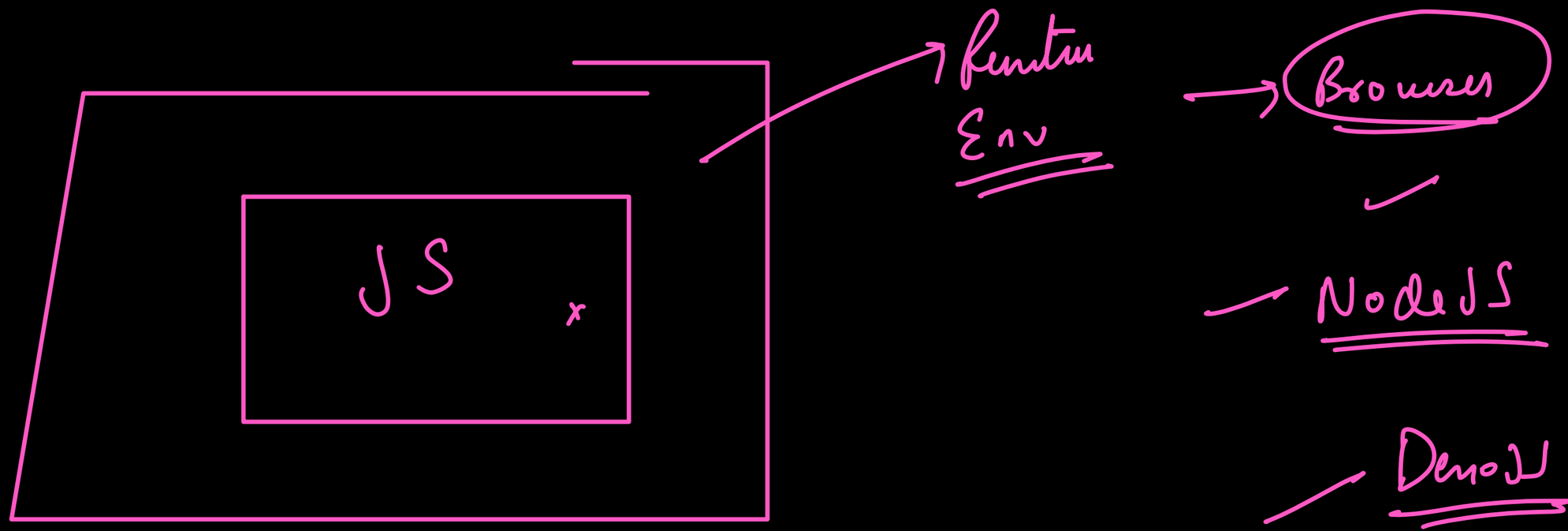
This is a block of while loop. As soon as we complete the last line of the block, we go back to the condition of while.

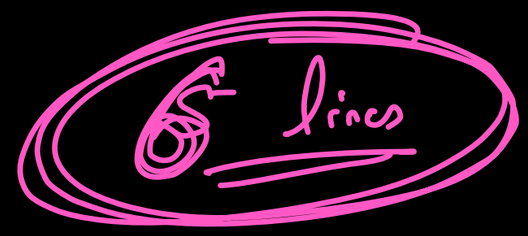
When to use a while loop or a for loop ??

Q: Print the first 20 natural no. in reverse order.



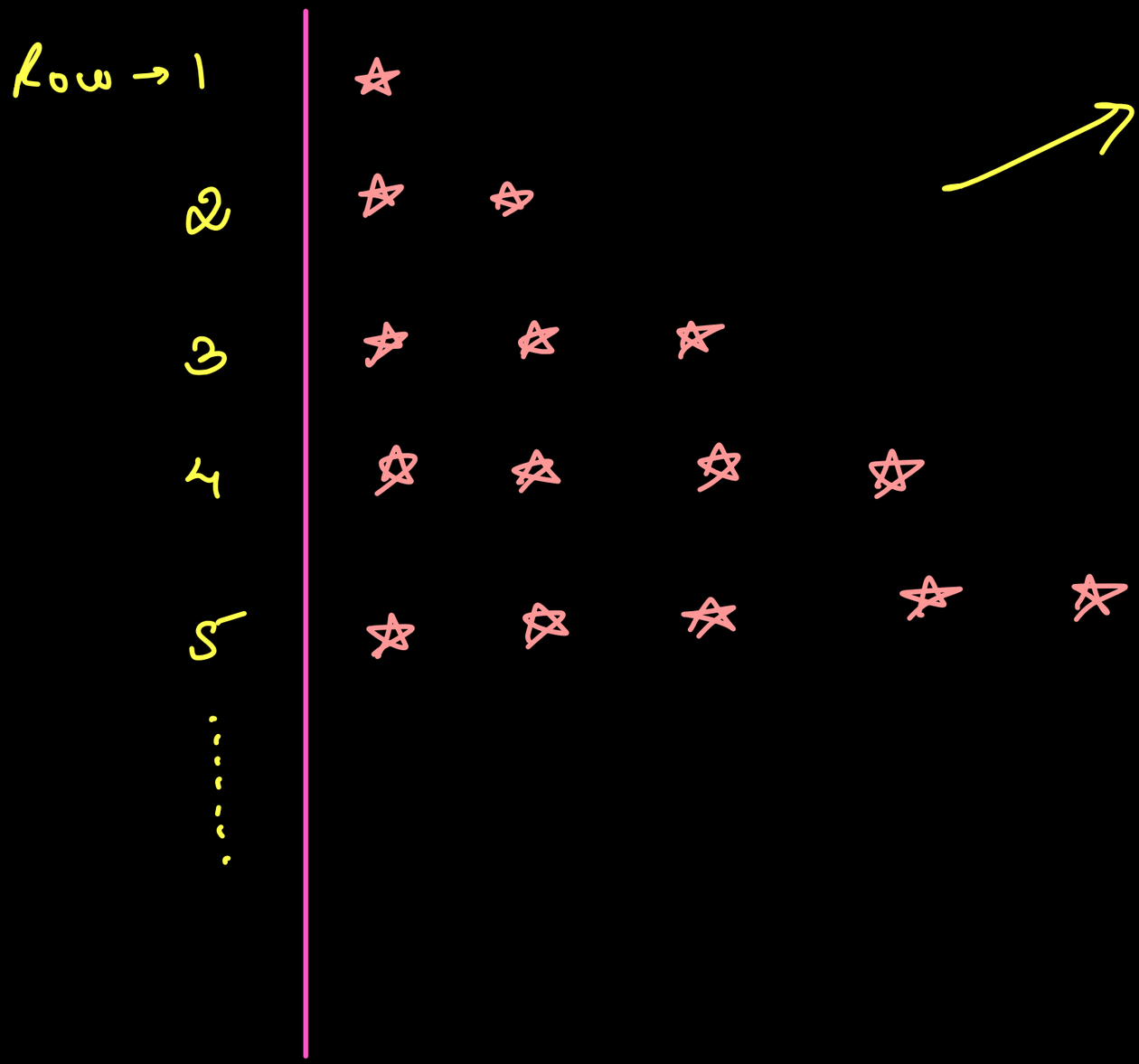
```
let i = 20;
while (i >= 1) {
    print(i)
    i = i - 1
}
```





Donot Hand code

JOIN THE DARKSIDE



(1) In any i^{th} row we print i no. of Stars in same line.

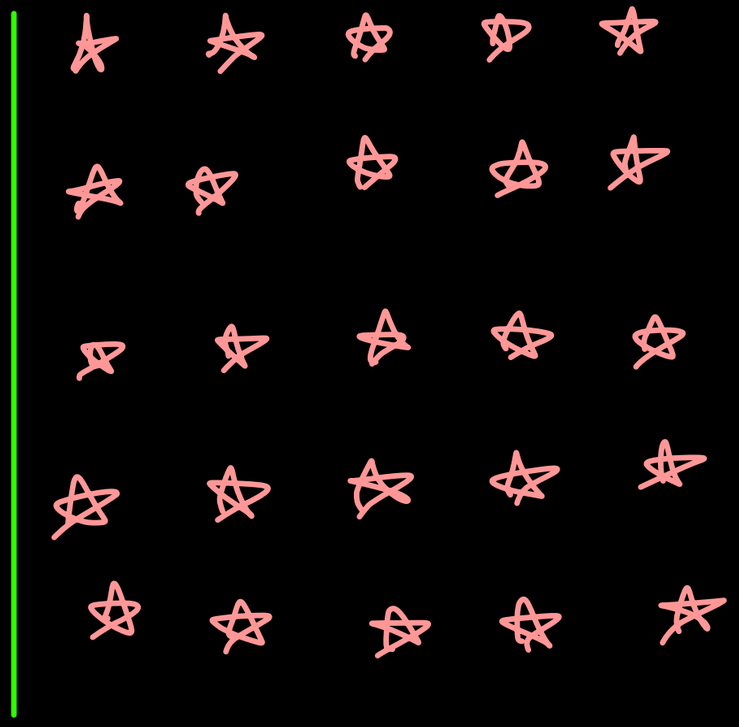
if somehow we can repeat this process

Given a value of i, print i stars in a single line.

```
for (let j = 1 ; j <= i ; j++)
        process.stdout.write("*");
}
```

→ if we repeat this algo for every row ??

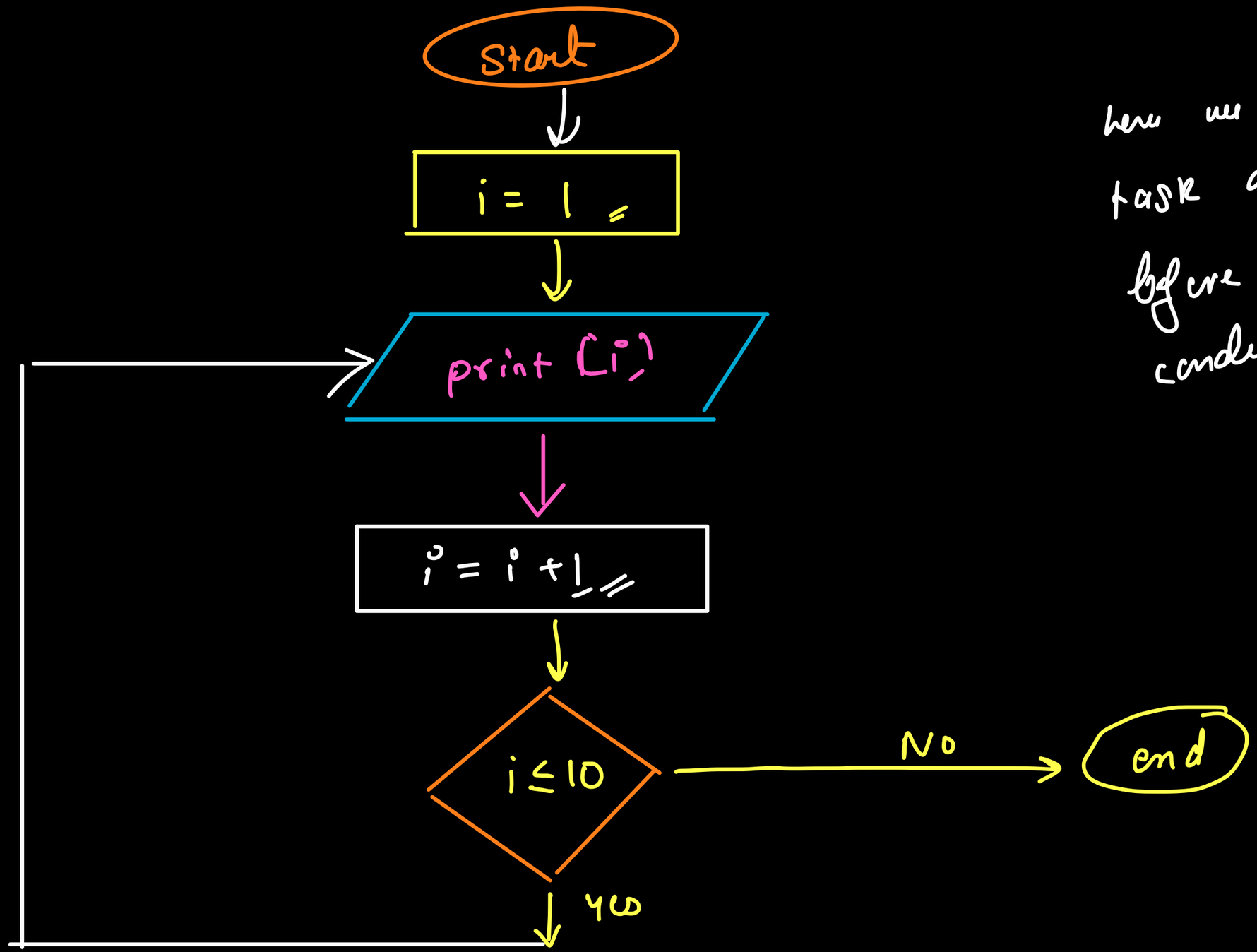
For a given value of n, print the following pattern



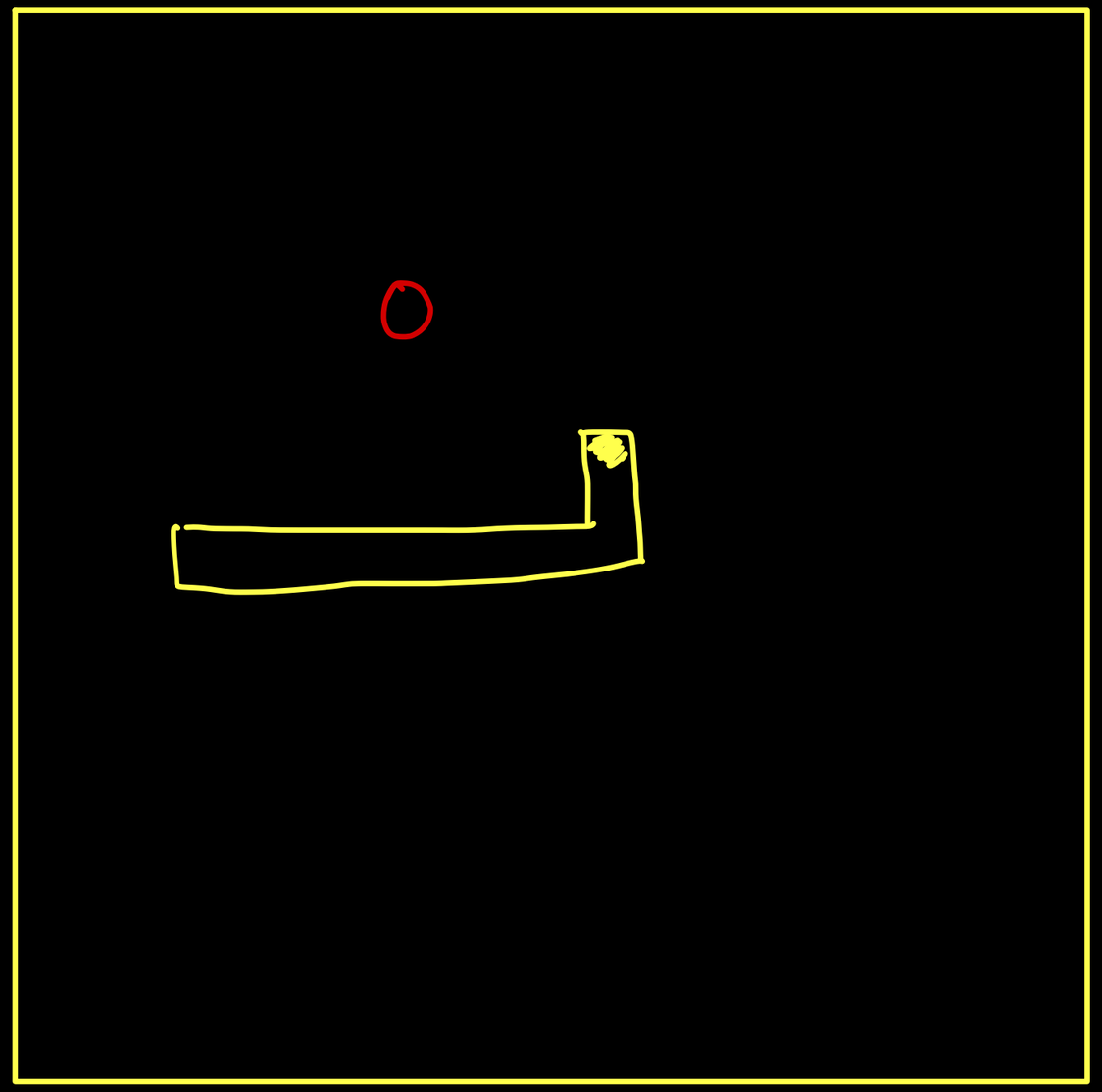
n = 5

for any ith row
how many stars ??

n stars



here we do the task allocation before checking condition



We want to gen food item randomly. If food is generated at same place where Snake body is then regenate food again.

initialize variable

```
do {
    ===
    ===
    ===
    ===
    increment
} while ( condition );
```

Q. Given a number x, return the sum of digits of x.

Ex → $x = 5213$

ans → 11

Ex → $x = 1003$

ans → 4

$x =$ 5 2 1 3 → Sum up the digits of a no.

↳ how to extract each digit.

5 2 1 3 ⇒ 5 2 1 × 10 + 3

↳ remainder

↓

divide 5 2 1 3 with 10

$10 \overline{)\,5213\,(521}$

5 2 1 0

3

→ In the quotient we obtained last digit.

That means if we want to extract the last digit of a no, we can get the remainder of division by 10.

$x = 5213$ ✓ x

$x \% 10 \rightarrow 5213 \% 10 \rightarrow 3$

I got the last digit now let's eliminate it.

$x = Math.floor(x/10) \rightarrow 521$

$521 \rightarrow 521 \% 10 \rightarrow 1$

$521/10 \rightarrow 52$

$52 \rightarrow 52 \% 10 \rightarrow 2$

$52 \rightarrow 52/10 \rightarrow 5$

S ⟶ S%10 ⟶ S

S ⟶ S/10 ⟶ 0

for (———) c

if (———) c

Continue;

}

```
for (  ——————  ) {
        .
        .
        .
        continue;

}
```

Switch → nothing to do with loops

↳ everything that Switch does is achievable by normal if else.

```
switch (expression) {
    case v1:
        ////
        break
    case v2:
        ////
        break
    .
    .
    .
    default:
        ////
}
```

JOIN THE DARKSIDE