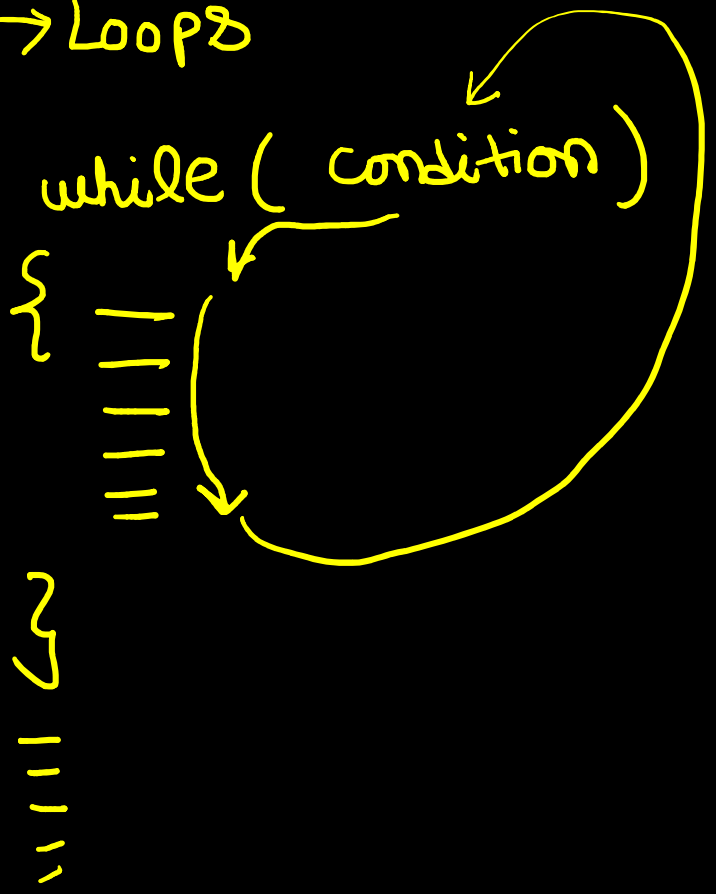
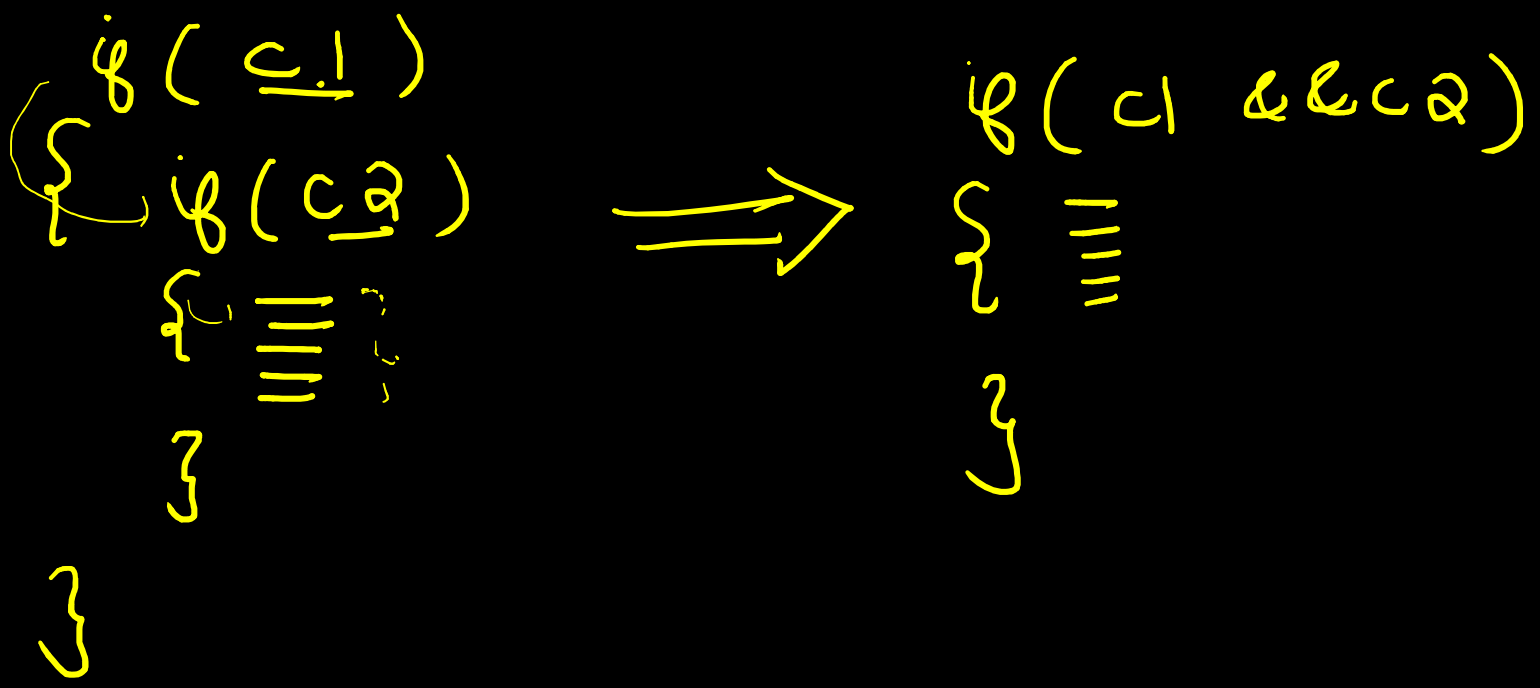
→ Loops



for(int i=0;i<n;i++)
{ ≡ }

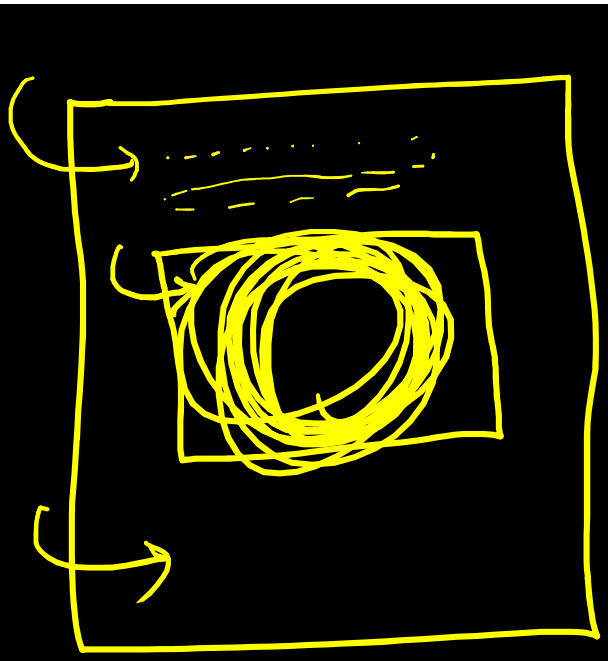
0 .... n-1


if(c1)
{
if(c2)
{
}
}
if(c1 && c2)
{
}

Nesting of Loops

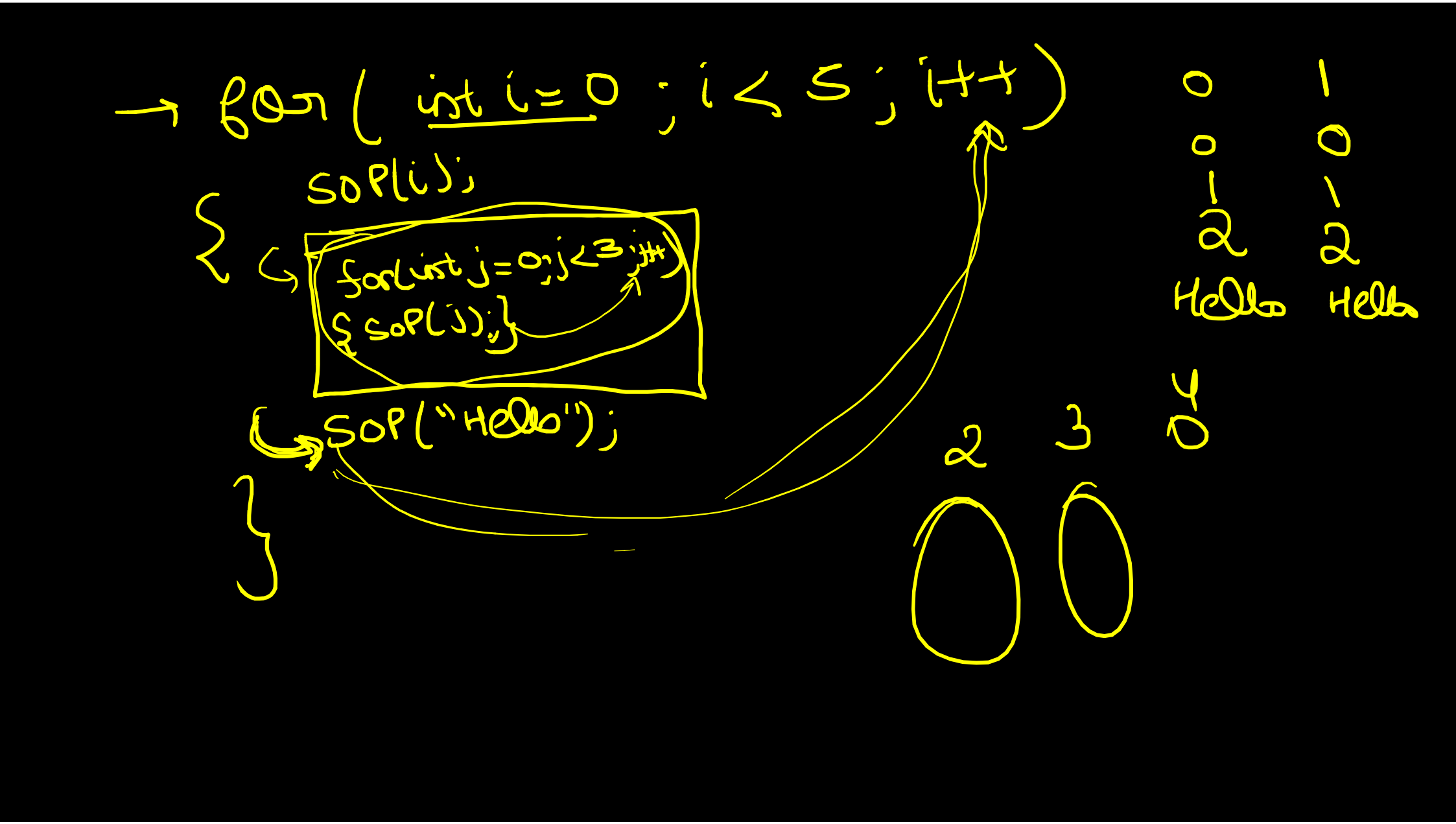
```
for ( int i=0; i<5 ; i++ )
{
    for( int j=0; j<4 ; j++)
    {
        sop ("Hello");
    }
}
```

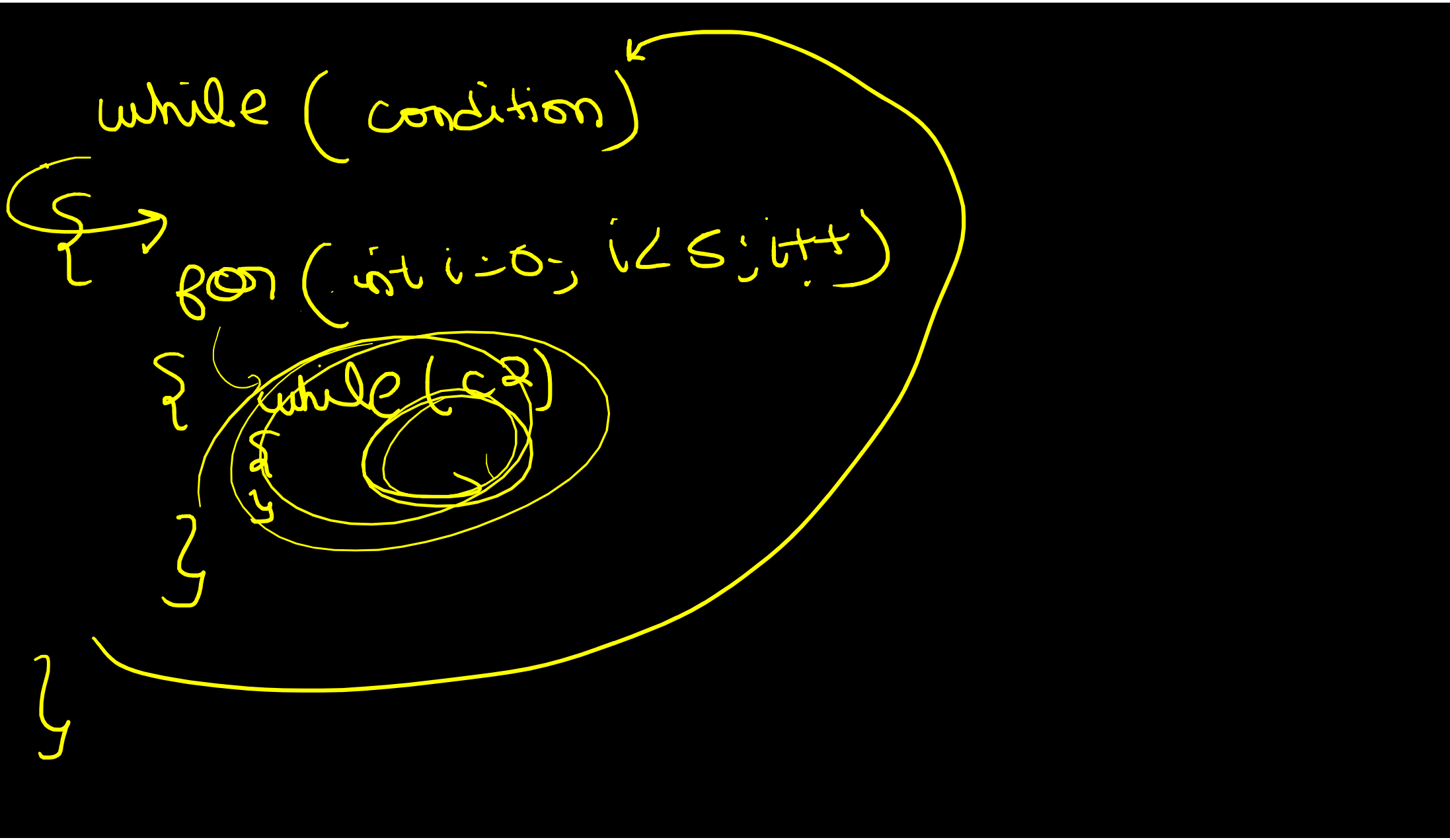
i = 0, j = 0 → Hello
→ Hello
i = 0, j = 1
j = 2 → Hello
j = 3 → Hello



```
for ( int i=0 ; i<5 ; i++ )
{
    SOP(i);
    for ( int j=0 ; j<4 ; j++ )
    { SOP(j); }
}
```

0	1	2	3	4
0	0	0	0	0
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3


for(int i=0 ; i<5 ; i++)
{
SOP(i);
for(int j=0;j<3;j++)
{SOP(j);}
SOP("Hello");
}
0 1
0 1
1 2
1 2
Hello Hello
2 3 4
0 0


while (condition)
{
for (int i=0; i<5; i++)
{
while (c2)
{
}
}
}

① * * * *

② * * * *

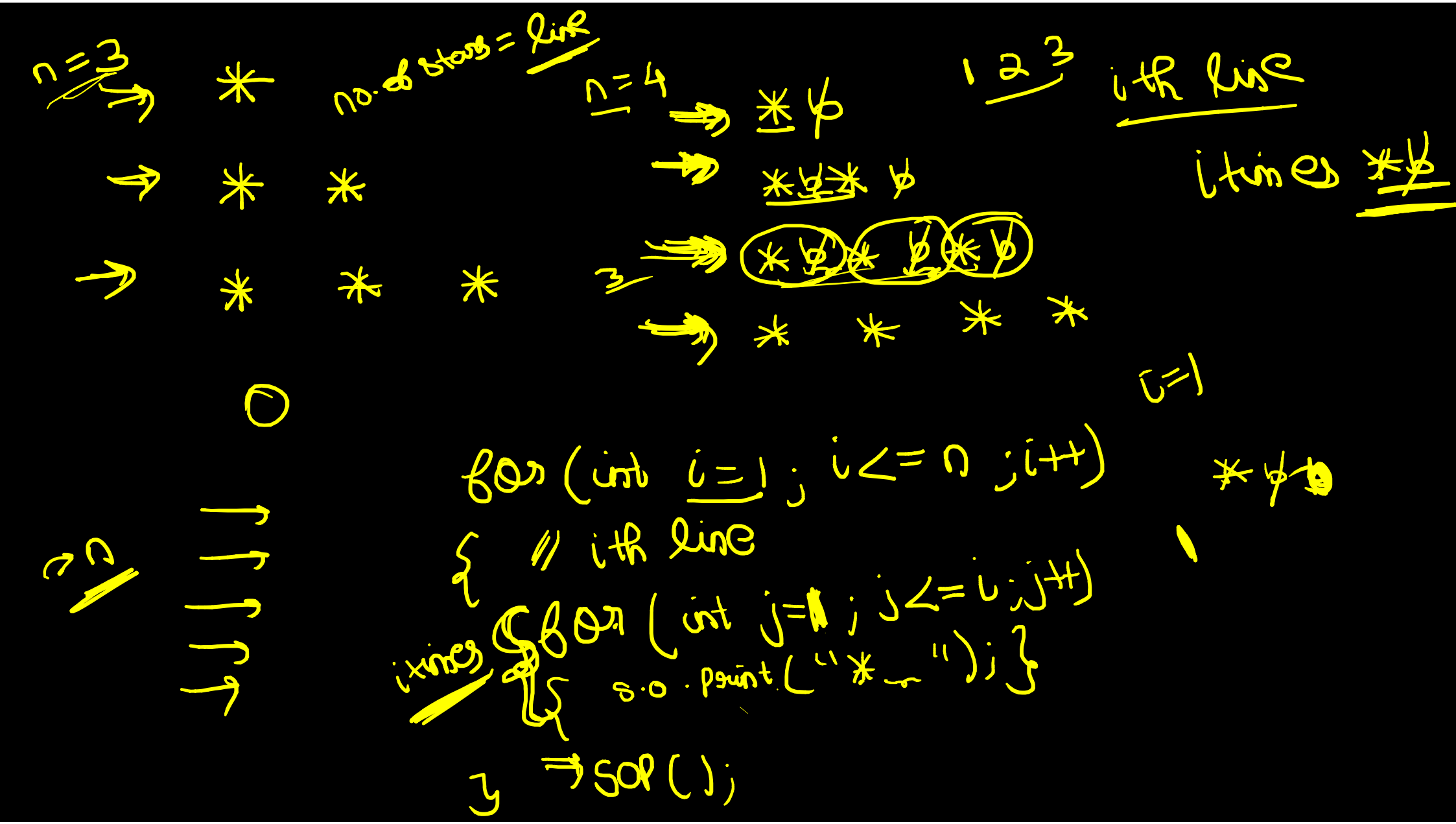
③ * * * *

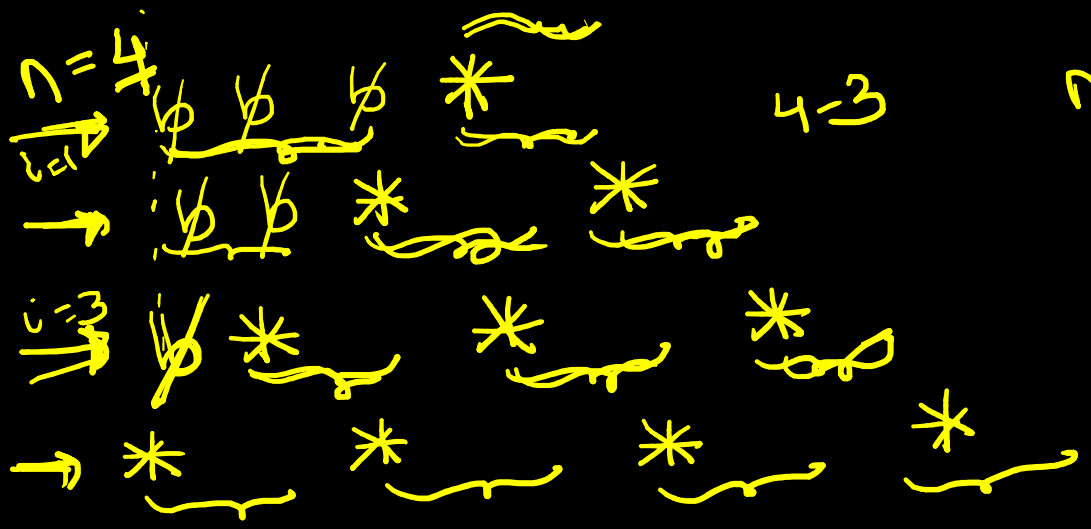
④ * * * *

⑤ * * * *

```
for(int i=1 ; i<=5 ; i++)
{  // ith line
   for(int j=1; j<=4 ; j++)
   {  S.O.Print("*");
   }
   S.O.P("")
}
```

```
****|
 ***|
  **|
   *|
```


n=3
no. of stars = line
n=4
1 2 3
ith line
itimes *
for (int i=1; i<=n ;i++)
{ // ith line
for (int j=1; j<=i;j++)
{ s.o.print("* ");}
⇒SOP();
i=1
*
n=2



ith line *␣

ith (n-i) spaces

4 → 0

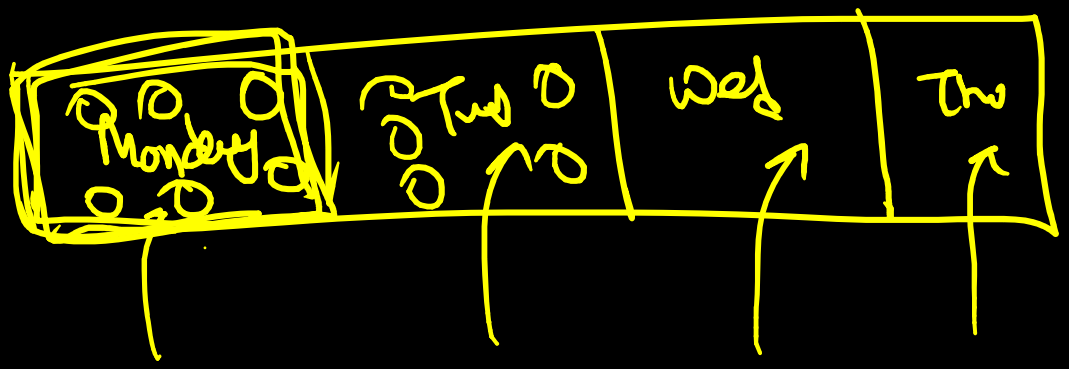
3 → 1

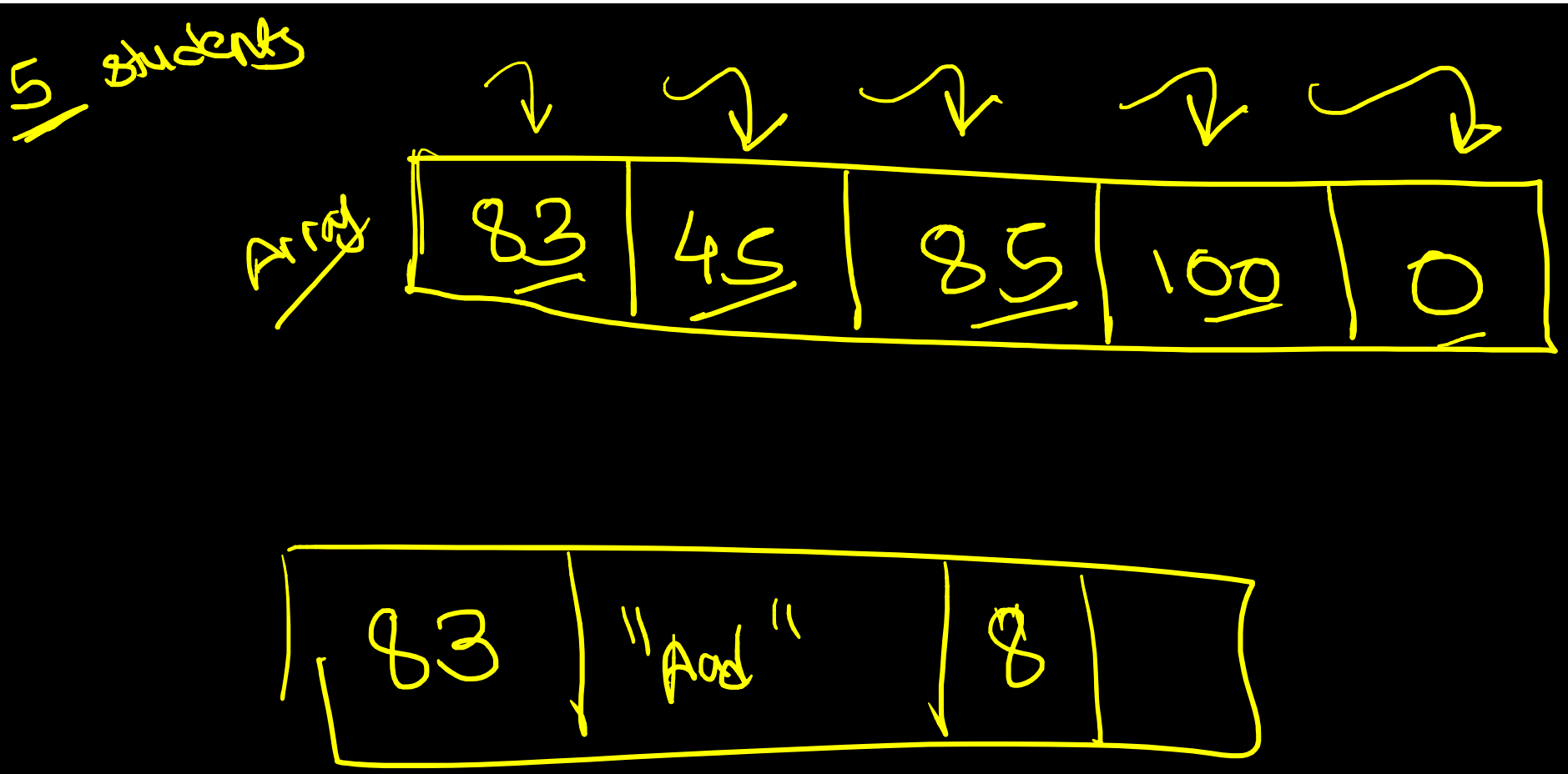
2 → 2

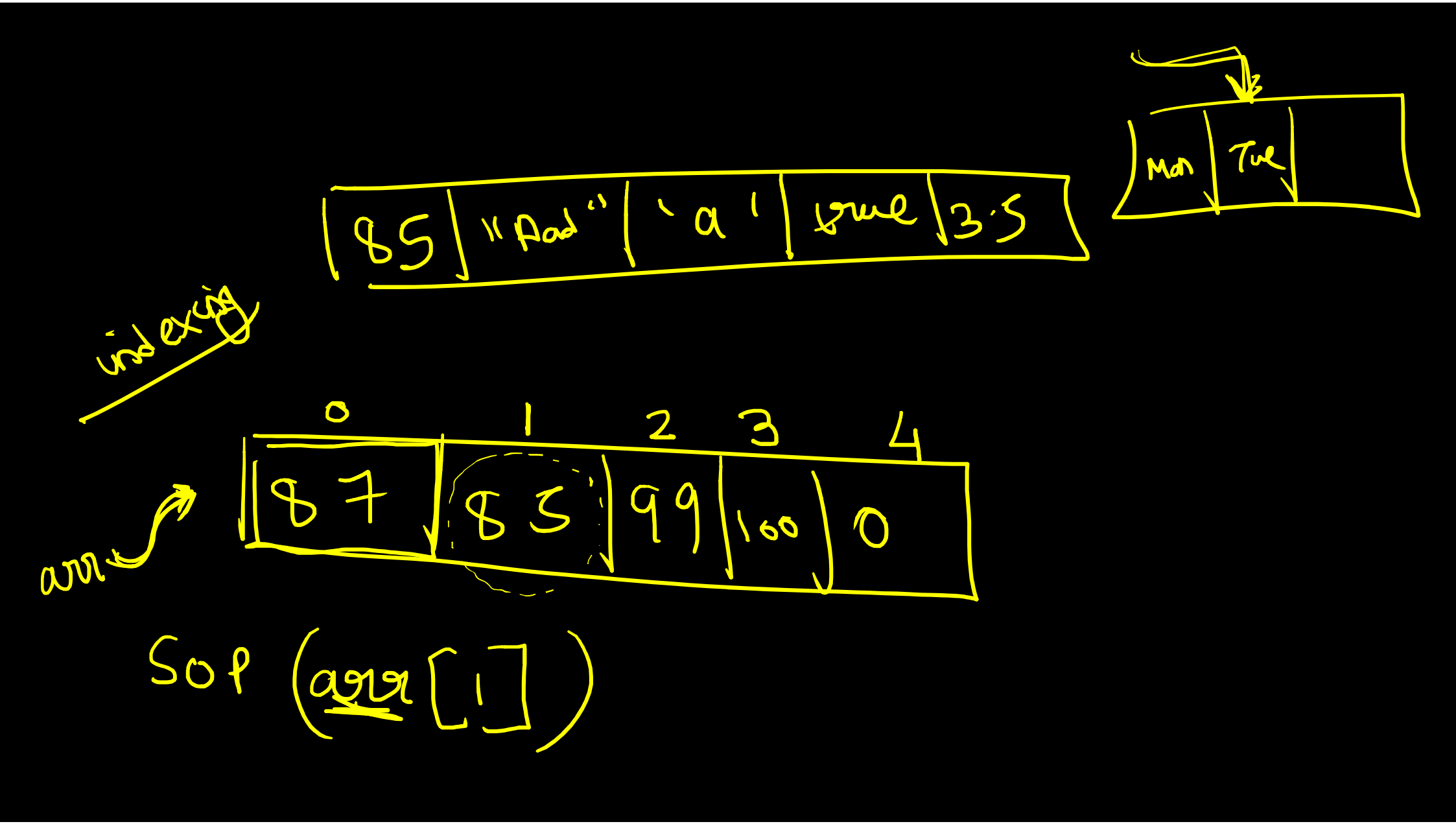
1 → 3

Arrays

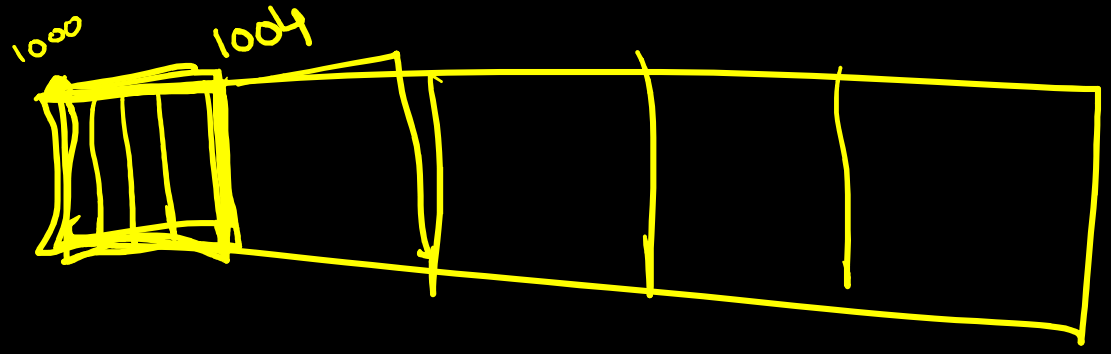
```
int marks = 100;
int marks2 = 50;
int marks3 =
```




5 students
Array
83
45
85
100
0
83
"Asd"
8

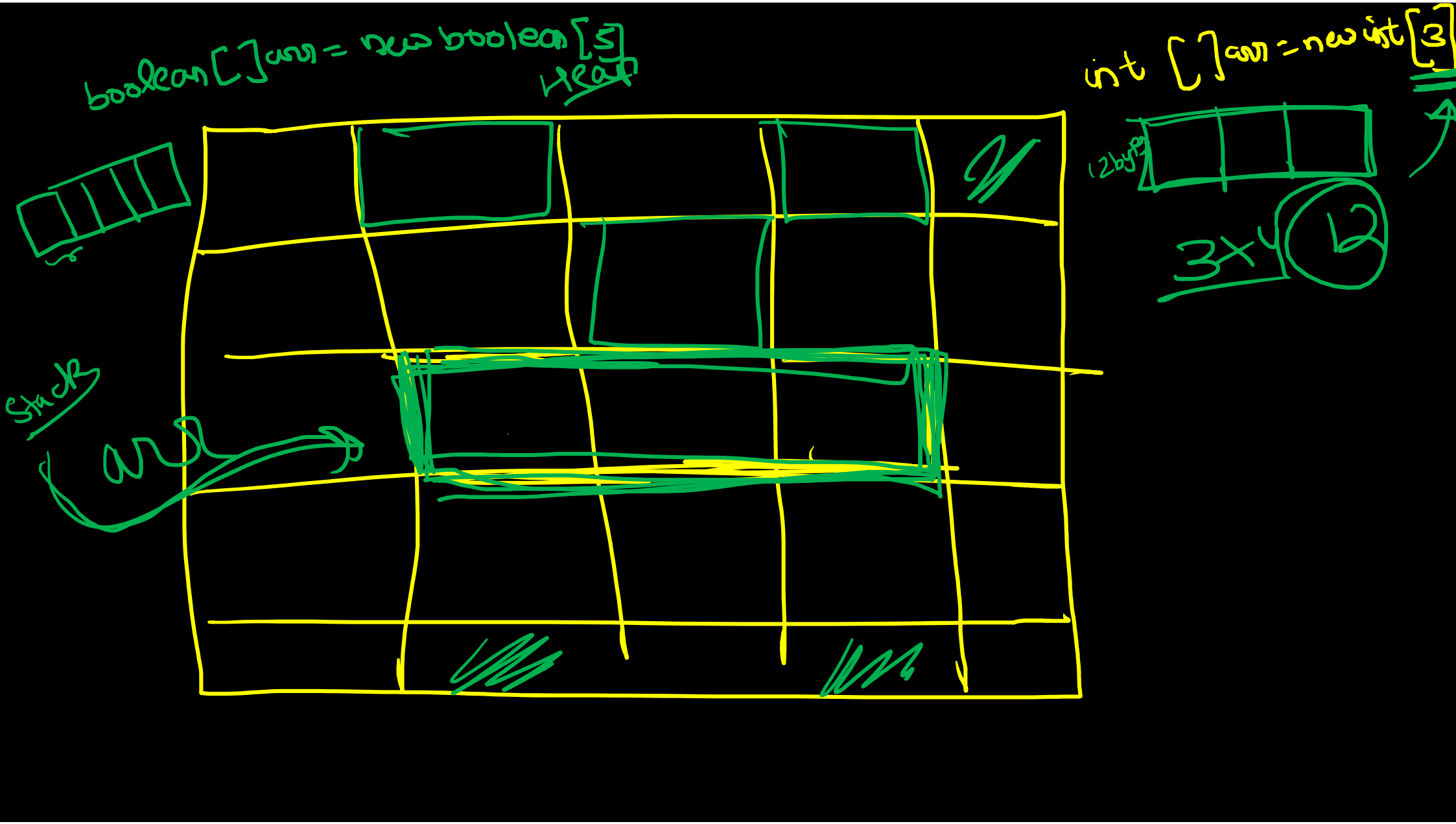

85
"Aad"
'a'
true
3.5
Mon
Tue
index
0
1
2
3
4
87
85
99
100
0
arr
Sop (arr[1])

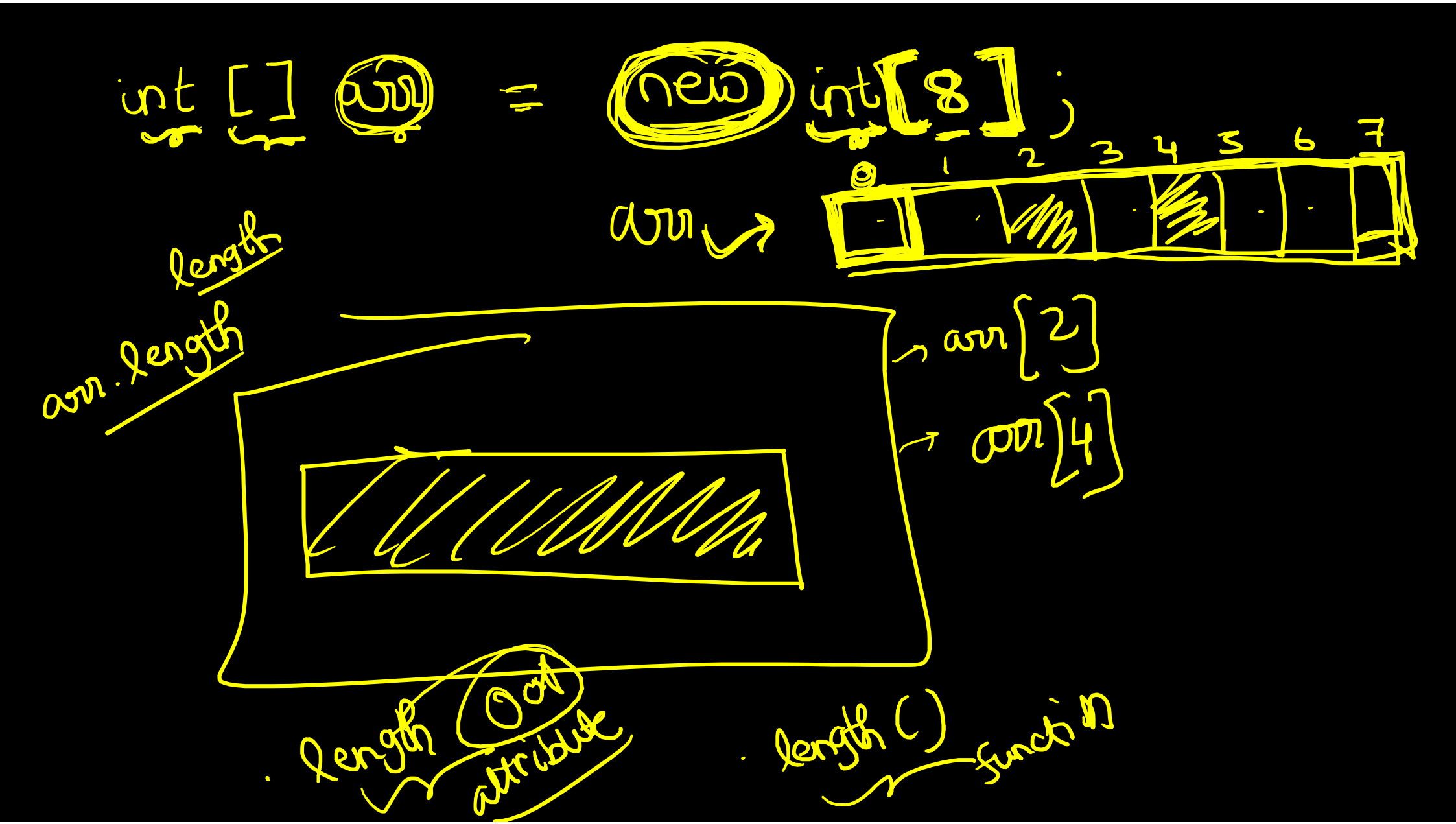
```
int [] arr = new int[5];
```



arr[]

0	1	2	3	4
85		55	44	33


boolean[] arr = new boolean[5]
Heap
int [] arr = new int[3]
12 bytes
3x4
12
Stack
arr


int [] arr = new int[8];
0 1 2 3 4 5 6 7
arr
length
arr.length
arr[2]
arr[4]
.length
Array attribute
.length()
function