

Given an integer n , you need to recreate the pattern given below for any value of N . Let's say for $N=3$, the pattern should look like as below:

```
*****
*****
*****
*****
Print the pattern in the function given to you.
```

Given an integer n , you need to recreate the pattern given below for any value of N . Let's say for $N=5$, the pattern should look like as below:

```
*****
*****
*****
*****
*****
Print the pattern in the function given to you.
```

Given an integer n , you need to increase the pattern given below for any value of N . Let's say for $N=5$, the pattern should look like as below:

```
10
100
1000
10000
100000
Print the pattern in the function given to you.
```

Given an integer n , you need to recreate the pattern given below for any value of N . Let's say for $N=5$, the pattern should look like as below:

```
1
22
333
4444
55555
Print the pattern in the function given to you.
```

Given an integer n , you need to increase the pattern given below for any value of N . Let's say for $N=5$, the pattern should look like as below:

```
100
1000
10000
100000
1000000
Print the pattern in the function given to you.
```

Given an integer n , you need to decrease the pattern given below for any value of N . Let's say for $N=5$, the pattern should look like as below:

```
*****
*****
*****
*****
*****
Print the pattern in the function given to you.
```

Given an integer n , you need to decrease the pattern given below for any value of N . Let's say for $N=5$, the pattern should look like as below:

```
*****
*****
*****
*****
*****
Print the pattern in the function given to you.
```

Given an integer n , you need to recreate the pattern given below for any value of N . Let's say for $N=5$, the pattern should look like as below:

```
*****
*****
*****
*****
*****
Print the pattern in the function given to you.
```

Given an integer n , you need to recreate the pattern given below for any value of N . Let's say for $N=5$, the pattern should look like as below:

```
*****
*****
*****
*****
*****
Print the pattern in the function given to you.
```

Given an integer n , you need to increase the pattern given below for any value of N . Let's say for $N=5$, the pattern should look like as below:

```
1
11
111
1111
11111
Print the pattern in the function given to you.
```

Given an integer n , you need to recreate the pattern given below for any value of N . Let's say for $N=5$, the pattern should look like as below:

```
1
11
111
1111
11111
Print the pattern in the function given to you.
```

Given an integer n , you need to recreate the pattern given below for any value of N . Let's say for $N=5$, the pattern should look like as below:

```
12
122
1222
12222
122222
Print the pattern in the function given to you.
```

8

pattern should look like as below:
 1
 12
 123
 1234
 123456789

Print the pattern in the function given to you.

```
if (N-2 == 0) {
    cout << 1;
}
else {
    cout << "0 ";
    while (j <= i) {
        cout << j;
        if (j != i) cout << " ";
        j++;
    }
    cout << endl;
}

Case 2 : i = 1
N-2 = 0;
j = 0;
if (j <= i) {
    cout << j;
    if (j != i) cout << " ";
    j++;
}
cout << endl;

Case 3 : i = 2
N-2 = 1;
j = 0;
if (j <= i) {
    cout << j;
    if (j != i) cout << " ";
    j++;
}
cout << endl;
```

Given an integer n. You need to generate the pattern given below for any value of N. Let's say for N = 5, the pattern should look like as below:
 1
 12
 123
 1234
 12345

Print the pattern in the function given to you.

Given an integer n. You need to generate the pattern given below for any value of N. Let's say for N = 5, the pattern should look like as below:
 1
 12
 123
 1234
 12345

Print the pattern in the function given to you.

Given an integer n. You need to generate the pattern given below for any value of N. Let's say for N = 5, the pattern should look like as below:
 A
 AB
 ABC
 ABCD
 ABCDE

Print the pattern in the function given to you.

Given an integer n. You need to generate the pattern given below for any value of N. Let's say for N = 5, the pattern should look like as below:
 ABCDE
 ABC
 ABC
 ABC
 A

Print the pattern in the function given to you.

```
int i=0;
while (i < 4) {
    int j=0;
    while (j < i) {
        cout << "X ";
        j++;
    }
    cout << endl;
    i++;
}
```

$i < 3$

```
public class pat {
    public static void main(String[] args) {
        int rows=1;
        while (rows < 4) {
            int cols=1;
            while (cols <= rows) {
                System.out.print("* ");
                cols++;
            }
            System.out.println();
            rows++;
        }
    }
}
```

$0 \rightarrow ④$

$i < 3$

```
int rows=1;
while (rows < 4) {
    int cols=1;
    while (cols <= rows) {
        System.out.print("* ");
        cols++;
    }
    System.out.println();
    rows++;
}
```

$① \rightarrow ②$

```
int rows=1;
while (rows < 4) {
    int cols=1;
    while (cols <= rows) {
        System.out.print("* ");
        cols++;
    }
    System.out.println();
    rows++;
}
```

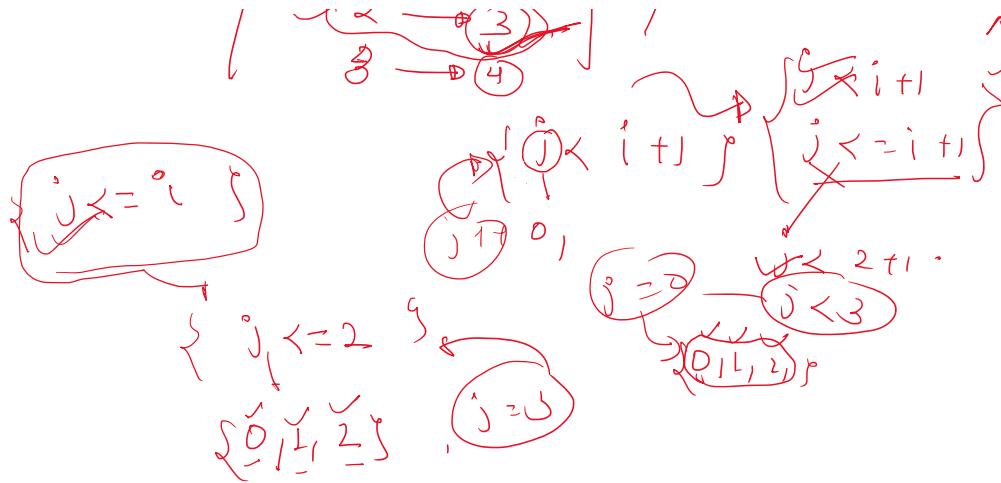
$② \rightarrow ③$

```
int rows=1;
while (rows < 4) {
    int cols=1;
    while (cols <= rows) {
        System.out.print("* ");
        cols++;
    }
    System.out.println();
    rows++;
}
```

$③ \rightarrow ④$

```
int rows=1;
while (rows < 4) {
    int cols=1;
    while (cols <= rows) {
        System.out.print("* ");
        cols++;
    }
    System.out.println();
    rows++;
}
```

$④ \rightarrow ⑤$





$\{$
 4 → 2
 5 → 1
 6 → 1

$\{$
 4 → 3
 5 → 2
 6 → 1

$\{$
 int i = 0;
 while ($i < 2$) {
 ↓
 $i / 2 \leq 2$
 ↓
 int j = 0;
 while ($j < i + 1$) {
 ↓
 print;
 ↓
 j++
 ↓
 }
 }
 }
 else {

$\{$
 4
 5
 6

$\{$
 4
 5
 6

$$\begin{aligned}
 (4) - (1 + 0) &= 3 \\
 5 - (2 + 1) &= 2 \\
 6 - (3 + 2) &= 1 \\
 \text{num} &= 1
 \end{aligned}$$

$$(4) - (i + \text{num})$$

$$\begin{aligned}
 \dots - (7 - 1) &= 3 \\
 (7 - 5) &= 2 \\
 (7 - 6) &= 1
 \end{aligned}$$

4
 5
 6

3
 2
 1

$$= \{ l - n + j \}$$

$$4 - (1 + 0)$$

$\{$
 6 → 2

$i = 1$
 $j < i - (j + \text{new})$

min or (*)

$j + t^1, h + t^1$

}

.....

, $n = L$

*)

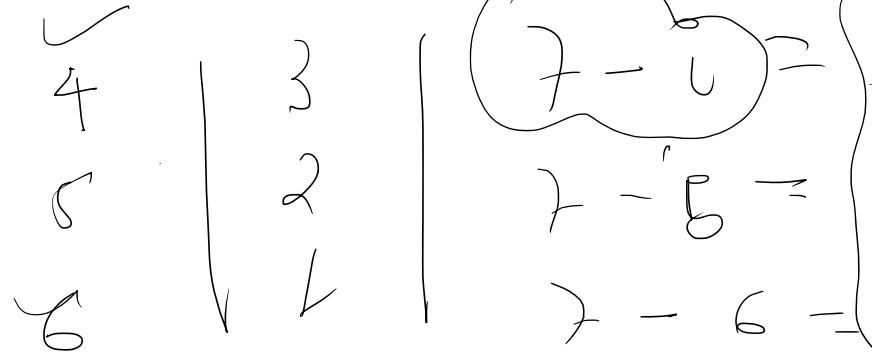
$j = 3$

$j \leq 8$

$t^1 = \{4 - z_t\}$

0	1
2	3
4	5
6	7

1	2	3
4	5	6
7	8	9
10	11	12



int i = 0;

if (n > mid = 7 / 2)

while (i < 7) {

if (i <= mid) {

int j = 0;

while (j < i + 1)

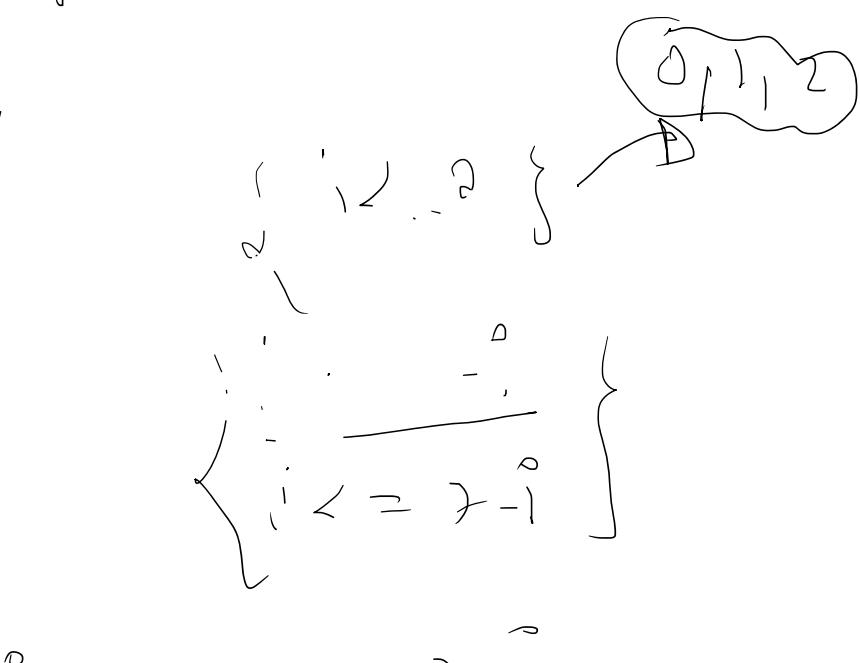
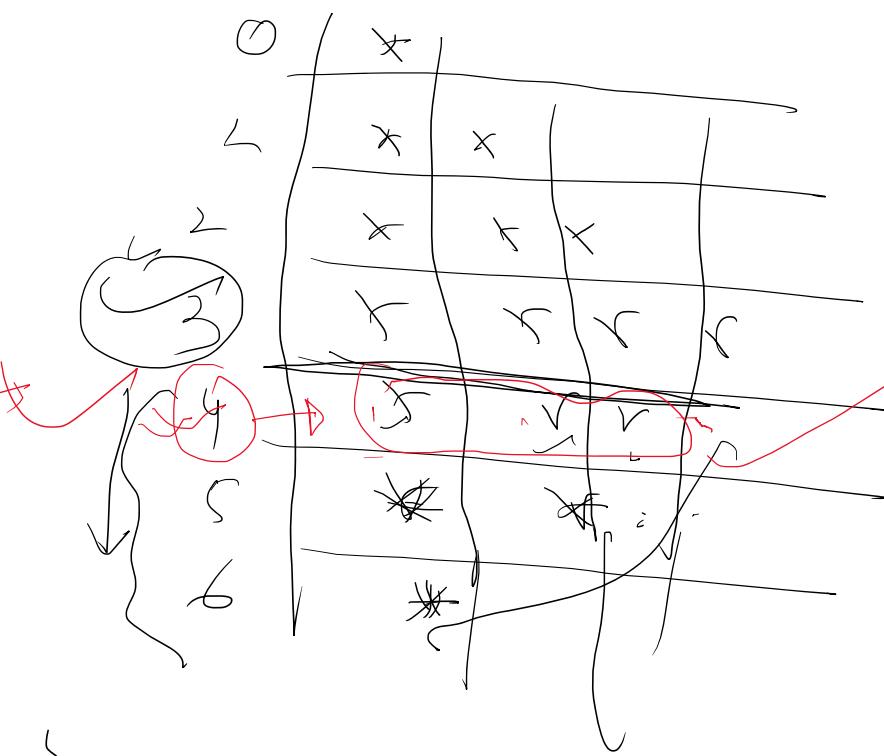
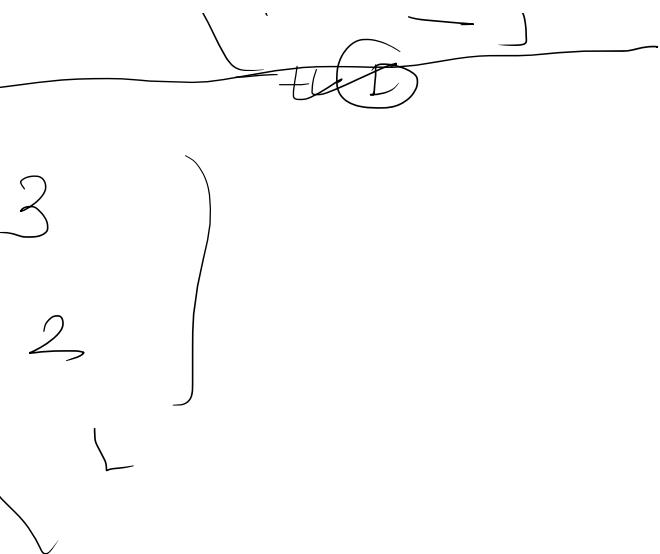
print(*);

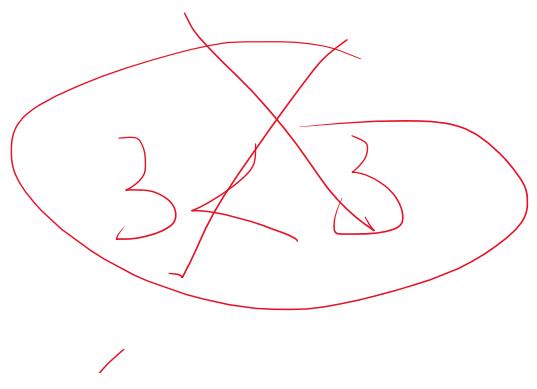
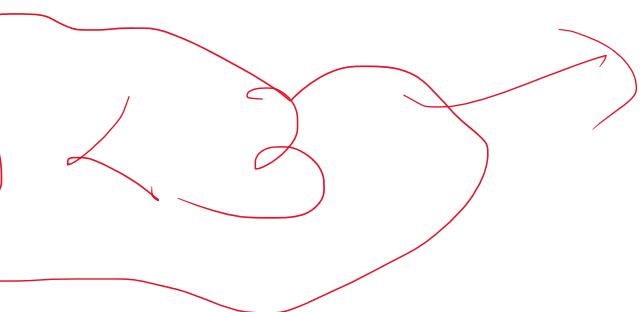
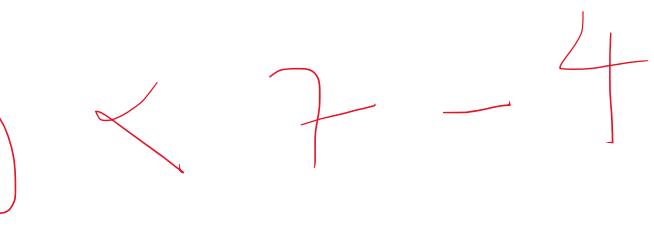
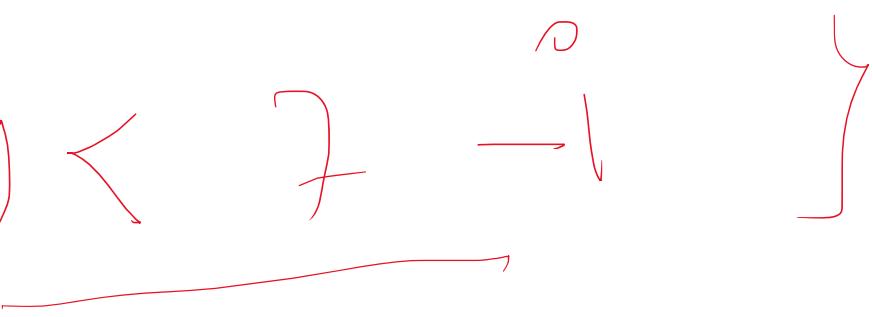
j++;

y

else {

int j = 0;





while ($j < 7$)

point(

i++)

}
}

point un()
^

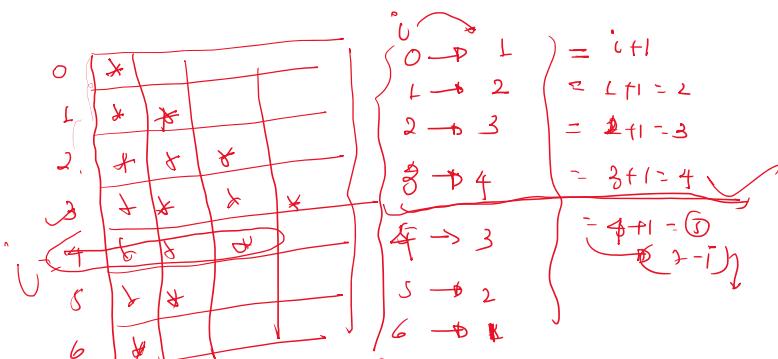
i++

int i = 0

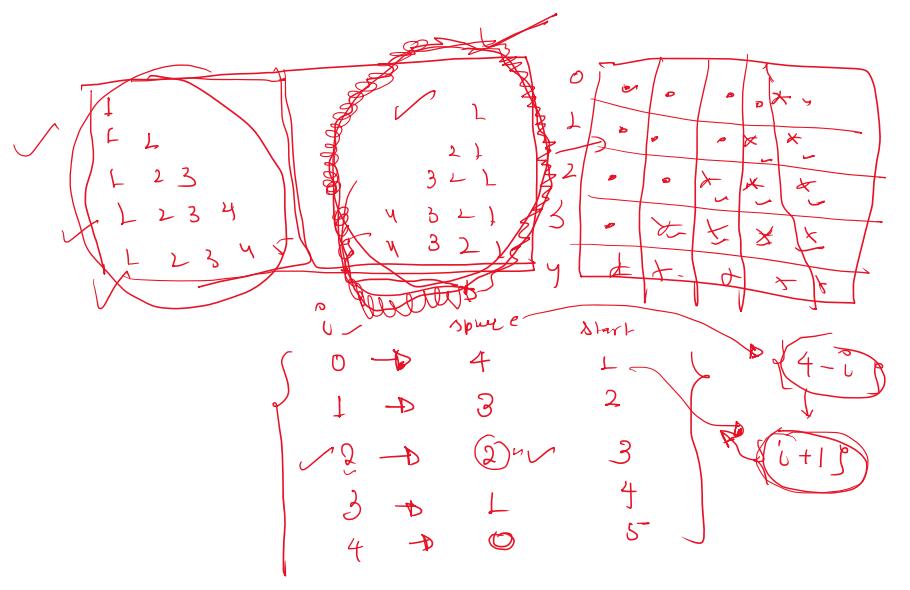
while (i <

if (i <=

int j = 0



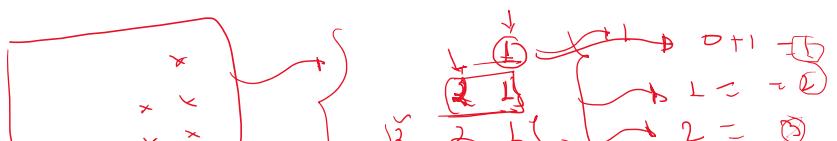
0
)
 S
)
 1
 ,
)
 2
 ,
)
 X
)
 1
 ,
)

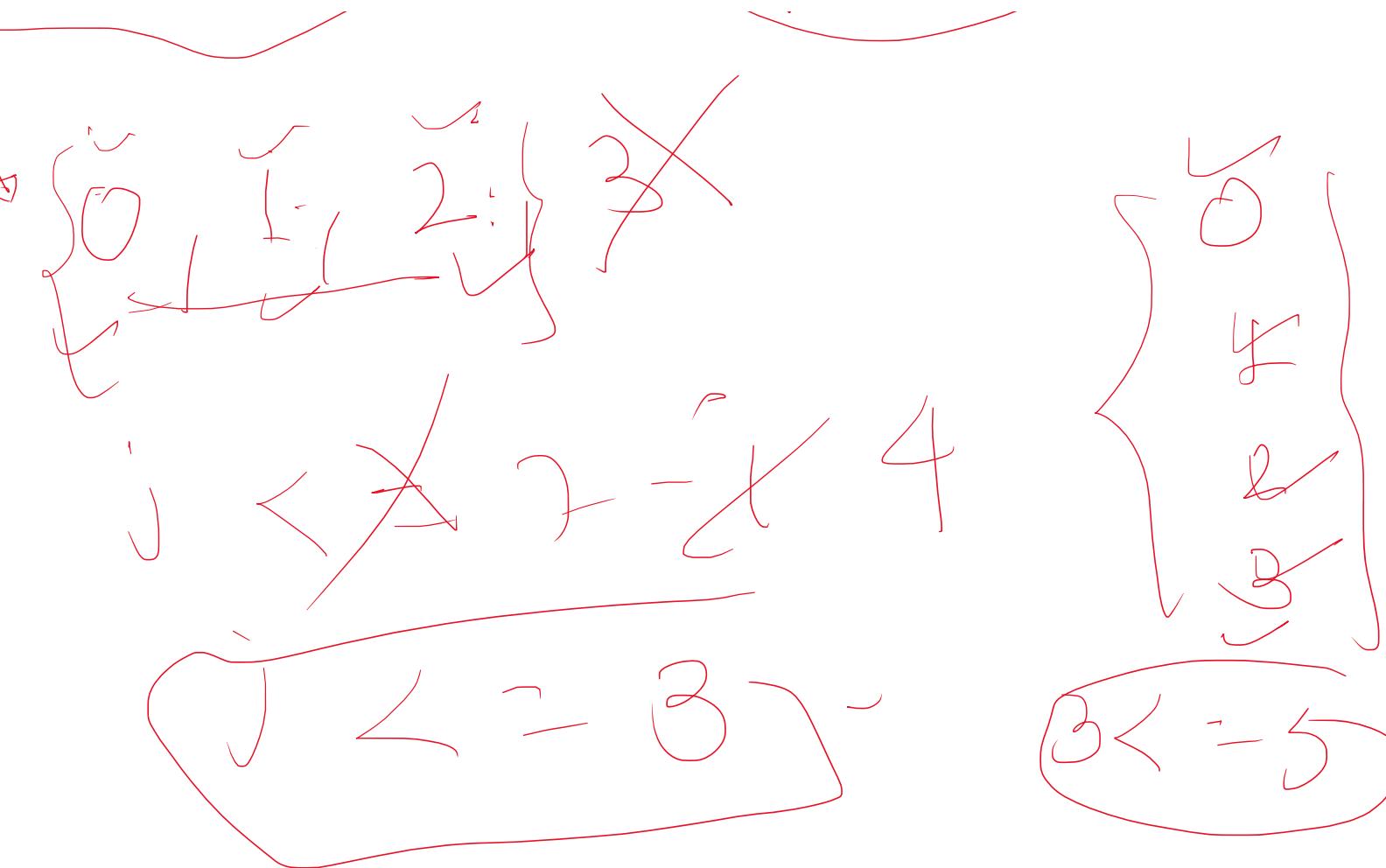


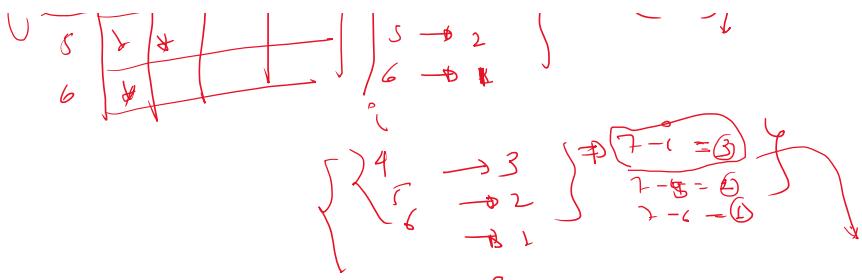
7) {

(7/2)) {

0 | \







$$\{ \begin{matrix} j < 7-i \\ i=0 \end{matrix} \} \quad \text{or} \quad \{ \begin{matrix} j \leq 7-i \\ i=1 \end{matrix} \}$$

$$\{ \begin{matrix} i=4 \\ j=4 \end{matrix} \}$$

$$\{ \begin{matrix} j \leq 7-i \\ i=2 \end{matrix} \}$$

$$\{ \begin{matrix} j < 7-i \\ i=3 \end{matrix} \}$$

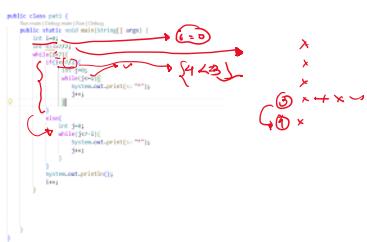
$$\{ \begin{matrix} j=0 \\ j=1, 2, 3 \end{matrix} \}$$

$$\{ \begin{matrix} j < 7-i \\ i=4 \end{matrix} \}$$

else {
 ...
}

int i;
while (i <

while (i <



for (y = 0; y < 5; y++)
 for (x = 0; x < 7 - y; x++)
 System.out.print("x");
 System.out.println();

for (y = 0; y < 5; y++)
 for (x = 0; x <

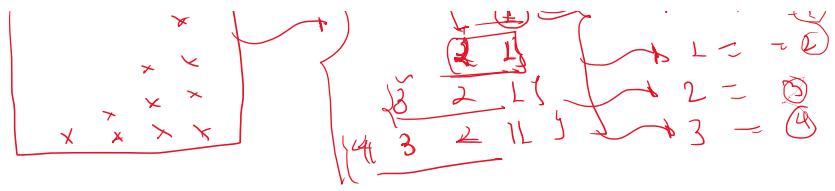
y
y

space $\Rightarrow \{ j < 3-i \}$
star $\Rightarrow \{ j < 2x + 1 \}$

```

int i=0
while(i<4){
    int j=0
    while(j<8-i){
        print("x");
        j++;
    }
}

```



$i < i + 1 \}$

$\eta (*)$

\dots

$= 0 \quad 1$

$\cup i < 7 - i \}$

$\eta (*)$
 $\hat{j} \neq \hat{i}$

$n()$

\vdash


```

int j=0;
while(j<8-i){
    print("  ");
    j++;
}

j=0;
while(j<2*i+1){
    print("*");
    j++;
}
print();
j++;

```

$i < 1$
 $\{ j < i \text{ for } j \leq i \}$
 $j \rightarrow 1, i \rightarrow 0$
 $j \rightarrow 2, i \rightarrow 0$
 $\text{if (start == 0) start = 1;}$
 else start = 0
 $\{ j < 2 \Rightarrow j \rightarrow start = 0 \}$
 else start = 1

