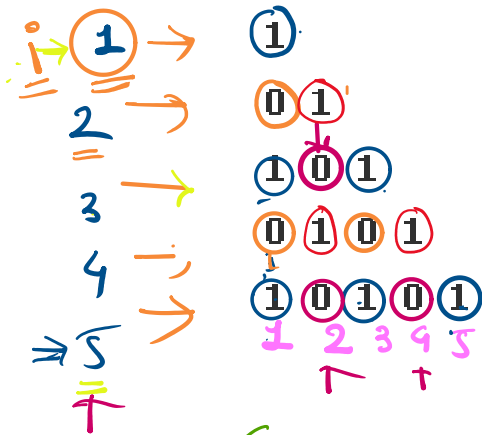


var =



1. Observation

2. Break it

a. Even iteration start with 0 and odd start with 1

b. If(i == even && j == odd) = 1

c. if(I == odd && j == even) = 0

d. If(i == even && j == odd) = 0

e. If(I == even && j == even) = 1

$$\text{if } (i = \text{odd} \quad j = \text{odd}) = \underline{\underline{1}}$$

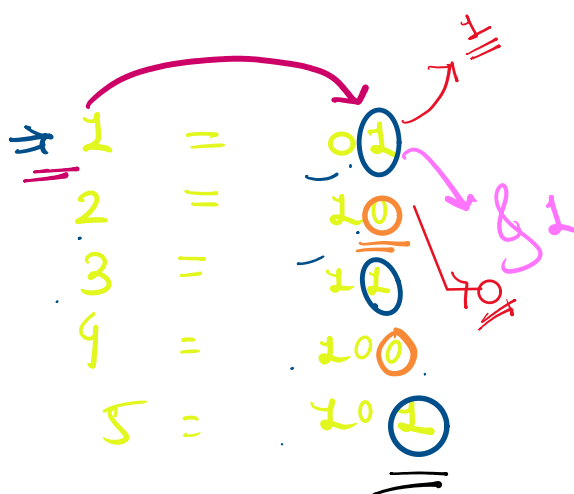
$$(\underline{\underline{i = \text{odd}}} \quad \&\& \quad j = \text{even}) = 0$$

$$\text{if } (i = \text{even} \quad j = \text{odd}) = 0$$

$$(i = \text{even} \quad \&\& \quad j = \text{even}) = 1$$

$$\text{if } (n \% 2 == 0) = \text{even}$$

$$\text{else odd}$$



$\&\& 1 \Rightarrow$ last binary digit

$$\begin{array}{r} 1 \\ \&\& 1 = 01 \\ \&\& 01 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 2 \&\& 1 = 10 \\ \&\& 01 \\ \hline 00 \end{array}$$

if (num % 2 == 0) {
 even

}
else {
 odd
}

```
new *
public class BinaryRightAngleTriangle {
    new *
    public static void main(String[] args) {
        int n = 5;
        for(int it = 1; it <= n; it++){
            for(int j = 1; j <= it; j++){
                int pIt = it & 1;
                int pJ = j & 1;
                if(pIt == 1 && pJ == 1) System.out.print(1 + " ");
                if(pIt == 1 && pJ == 0) System.out.print(0 + " ");
                if(pIt == 0 && pJ == 1) System.out.print(0 + " ");
                if(pIt == 0 && pJ == 0) System.out.print(1 + " ");
            }
            System.out.println();
        }
    }
}
```

A		B	
T	&	T	= T
T	&	F	= F
F	&	T	= F
F	&	F	= F

1 0 0	1 1 1
0 1 1	0 1 1
0 0 0	0 1 1

1 0 1	1 0 0
0 0 1	0 0 1
0 0 1 = 1	0 0 0 = 0

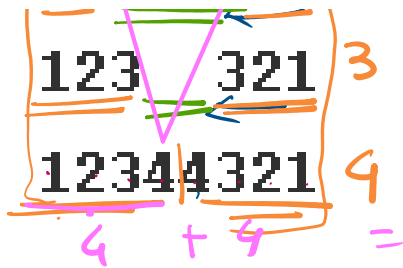
```
new *
public class BinaryRightAngleTriangle {
    new *
    public static void main(String[] args) {
        int n = 5;
        for(int it = 1; it <= n; it++){
            int var = it & 1;
            for(int j = 1; j <= it; j++){
                System.out.print(var + " ");
                var = var == 1 ? 0 : 1;
            }
            System.out.println();
        }
    }
}
```

1 0	var = 0
0 0 1	var = 0
0 0	var = 0

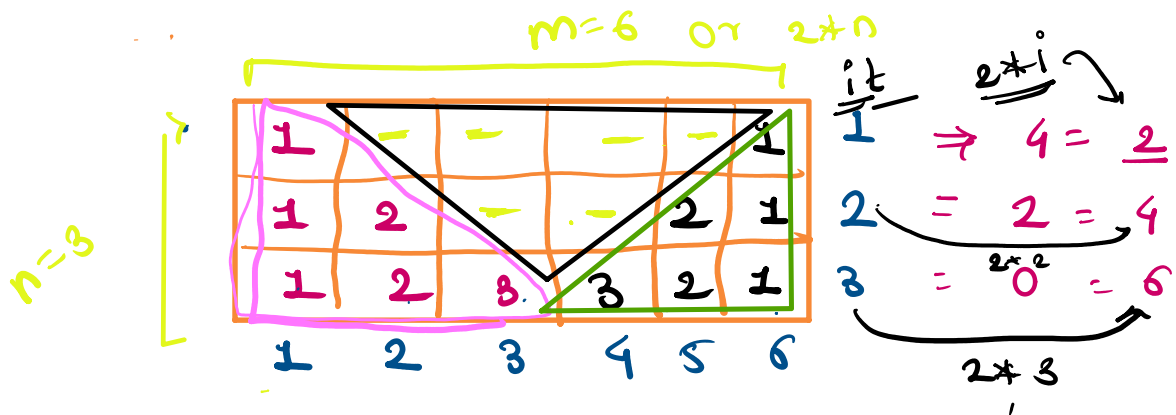
2) →

1	1	1
1 2	2 1	2
1 2 3	3 2 1	3

1. Observations →
2. Break it →
 - a. Print RAT one side both side



2×0 Cols



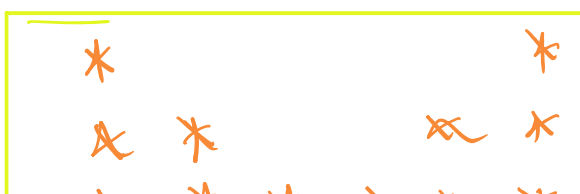
$n - 2 \times i \Rightarrow \text{space}$

Rule 3 -> nested Loop always run in rectangle format or square

```

public class MirrorRAT {
    new *
    public static void main(String[] args) {
        int n = 4;
        for(int it = 1; it <= n; it++){
            for(int num = 1; num <= it; num++){
                System.out.print(num + " ");
            }
            m = 2*n - 2*it
            for(int space = 1; space <= 2 * n - 2 * it; space++){
                System.out.print(" ");
            }
            for(int num = it; num >= 1; num--){
                System.out.print(num + " ");
            }
            System.out.println();
        }
    }
}

```



```

A *           x x *
x x x x x x x

```

```

* * * *
x           x

```

= 60 |

```

      *
     * *
    * * *
   * * *
  * * *
 * * *

```

```

*
*
*

```

```

- - *
- - *
x x x

```

```

1  * - - - *
2  x x - - x
3  x x x x x
4

```

$2 * 1$

$2 * 1 = 2$

$2 * 2 = 4$

$2 * 3 = 6$

$m = 2 * n$

$2 * n = 6$ ↓ (space)

$= 6 - 2 = 4$

$= 6 - 4 = 2$

$6 - 6 = 0$

X

```

  1
 2 1
3 2 1

```

✓

```

  1
 2 1
3 2 1

```

* num

```

*
*
*

```

Homework

A
AB
ABC
ABCD
ABCDE

ABCDE
ABCD
ABC
AB
A

A
BB
CCC
DDDD
EEEE