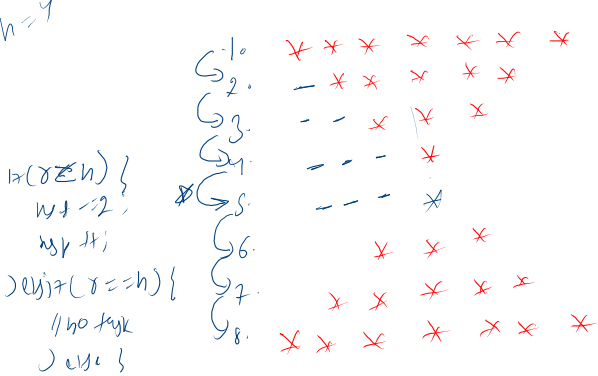
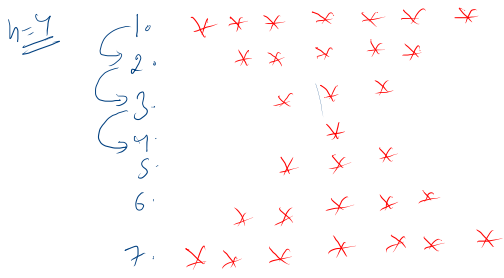


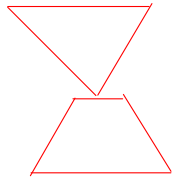
```
if(r<=h){
    nsp -=2;
    nsp ++;
}else if(r==h){
    // 마지막
}else{
    nsp+=2
    nsp--;
}
```

```
int nsp=1, nsp=7
```



```
if(r<h){   // r: 1,2,3
    nsp -=2;
    nsp ++;
}else{
    nsp +=2
    nsp--;
}
```

// r: 4,5,6,7



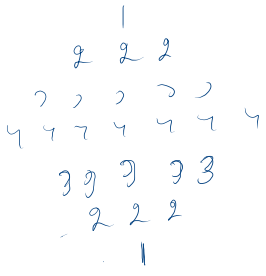
```
int nsp = 2n-1, nsp = 0;
for(int r=1; r<2*n; r++){
    for(space)
    for(star)
if(r<h){
    nsp -=2;
    nsp ++;
}else{
    nsp +=2
    nsp--;
}
```

n = 5

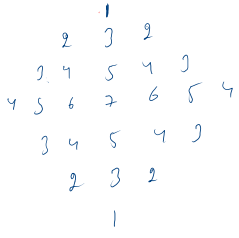
condition: (n is always odd number)

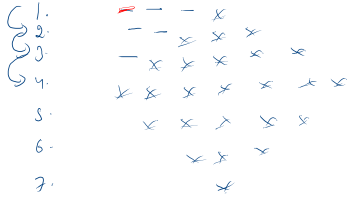
```
int nst = 1, nsp = (n/2)
for (int r = 1; r <= n; r++) {
    for (spaces)
    for (stars)

if (r <= n/2) {        // r : 1, 2, 3
    nst += 2;
    nsp--;
}
else {
    nst -= 2;
    nsp++;
}
```



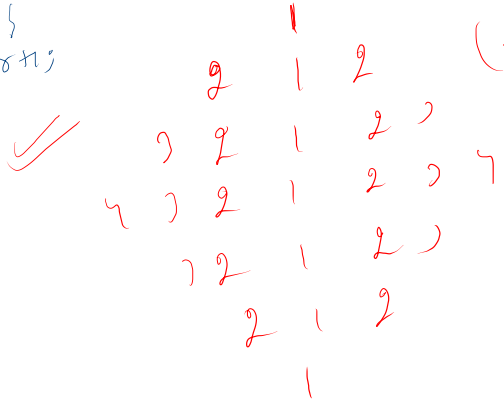
```
if (r >= n/2 + 1) {
    count = n - r + 1;
}
```





n	nst	nsp
5	1	2
7	1	3
9	1	4
11	1	5
n	1	(n/2)

(pattern thinking)



```java
int nst = 1, nsp = n / 2;
for (int r = 1; r <= n; r++) {
    for (int csp = 1; csp <= nsp; csp++)
        System.out.print(s: "  ");

    int count = r;
    if (r > n / 2 + 1)
        count = n - r + 1;

    for (int cst = 1; cst <= nst; cst++){
        System.out.print(count + " ");
        if(cst <= nst / 2){
            count++;
        }else{
            count--;
        }
    }

    if (r <= n / 2) {
        nst += 2;
        nsp--;
    } else {
        nst -= 2;
        nsp++;
    }
    System.out.println();
}
```