Q:
```
for(int i=1; i<=10; i+=2){
    SOP(i)
}
```
Output: 1, 3, 5, 7, 9

Q: Loop to print * n times

```
for(int i=1; i<=N; i++){
    SOP("*")
}
```

i=1; i<N ⇒ [1, N-1] ⇒ N-1 times

i=0; i<N ⇒ [0, N-1] ⇒ N times

Q: Given N. Print the following pattern.

N=3

N=5

```
for(int row=1; row<=N; row++){
    for(int col=1; col<=N; col++){
        SOP("*");
```

```
    }
    SOPln();
}
```

Q: Given N, M. Print the following pattern.

N = 3, M = 2

N = 5, M = 4

```
for(int row = 1; row <= N; row++) {
    for(int col = 1; col <= M; col++) {
        SOP("* ");
    }
    SOPln();
}
```

(3,2)

N	M	
1	1	
1	2	
1	3	→ Break
2	1	
2	2	
2	3	→ Break
3	1	
3	2	

Output

```
* * ↵
* * ↵
* * ↵
```

3 → 3 → Break

4 → Break

Q: Given N. Print the following pattern.

N = 3

N = 5

Row	Stars
1	1
2	2
3	3
4	4
5	5

(N = 5)

```
for(int row = 1; row <= N; row++) {
    for(int col = 1; col <= row; col++) {
        SOP("* ");
    }
    SOPln();
}
```

(N = 3)

Row	Col	Output
1	1	* ↵
1	2 → Break	* * ↵

2	1
2	2
2	3 → Break
3	1
3	2
3	3
3	4 → Break

* * * ↵

Q: Given N. Print the following pattern.

N = 3

N = 5

N = 5
R = 4

Stars = N + 1 − R
= 5 + 1 − 4 = 2

(N = 5)

Rows		Stars	
1	+	5	6
2	+	4	6
3	+	3	6
4	+	2	6
5	+	1	6

(N = 3)

R		S	
1	+	3	4
2	+	2	4
3	+	1	4

row + stars = N + 1

stars = N + 1 − row

```
for(int row = 1; row <= N; row++) {
    for(int col = 1; col <= N+1-row; col++) {
```

```
        SOP("* ");
    }
    SOPLN();
}
```

Q: Given N. Print the following pattern.

N = 4

```
*
* 2
* 2 *
* 2 * 4
```

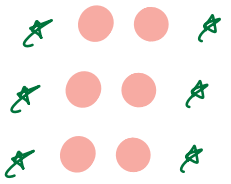
N = 5

```
*
* 2
* 2 *
* 2 * 4
* 2 * 4 *
```

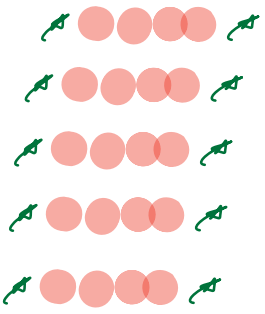
```
for(int row = 1; row <= N; row++) {
    for(int col = 1; col <= row; col++) {
        if(col%2 == 0) {
            SOP(col);
        } else {
            SOP("*");
        }
    }
    SOPLN();
}
```

Break 10:15

Q: N = 3



N = 5



⇒ space

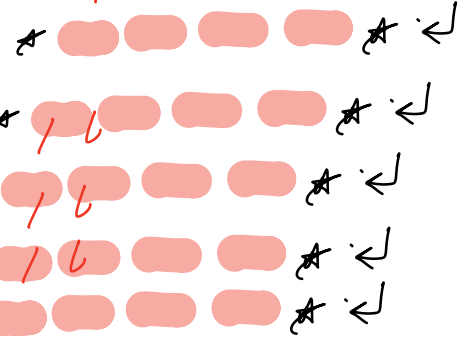
★ N-1 spaces ★

```
for(int row=1; row <= N; row++){

    SOP("*");
    for(int sp=1; sp<=N-1; sp++){
        SOP(" ");
    }
    SOP("*");
    SOPln();
}
```

(N=5)

Row	Col
1	★
1	1
1	2
1	3
1	4

Output:

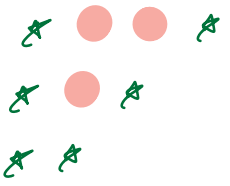


1 5 → Break

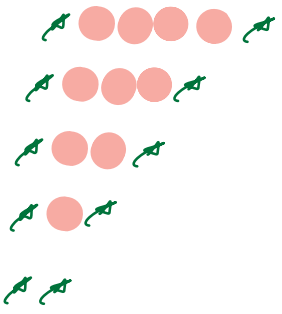
1 *

1 SOPln()

Q: N = 3



N = 5



Row		Spaces	(N = 5)
1	+	4	= 5
2	+	3	= 5
3	+	2	= 5
4	+	1	= 5
5	+	0	= 5

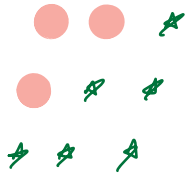
row + spaces = N

spaces = N - row

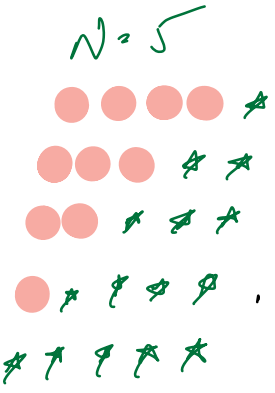
```
for(int row=1; row <= N; row++){

    SOP("*");
    for(int sp=1; sp <= N-row; sp++){
        SOP(" ");
    }
    SOP("*");
    SOPln();
}
```

Q: Given N. Print the following.

N = 3



N = 5



Row	Spaces	Stars	(N = 5)
1	4	1	
2	3	2	
3	2	3	stars = row
4	1	4	row + spaces = N
5	0	5	spaces = N - row

```
for(int row=1; row<=N; row++){

    for(int sp=1; sp<=N-row; sp++){
        SOP(" ");
    }
    for(int st=1; st<=row; st++){
        SOP("*");
    }
    SOPln()
}
```

Refresher ⇒ 1 month, experienced

Complete ⇒ 2 month, no experience

support@scaler.com

N = 143

Armstrong = N = sum of cube of digits of N

$143 = 1^3 + 4^3 + 3^3$

$= 1 + 64 + 27$

$\neq 92$

$153 = 1^3 + 5^3 + 3^3$

$= 1 + 125 + 27$

$= 153$