1. Given N, print :

N = 3

```
*
* 2
* 2 *
```

N = 5

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

N = 6

```
*
* 2
* 2 *
* 2 * 4
* 2 * 4 *
* 2 * 4 * 6
```

cols

rows	1	2	3	4	5	6
1	*					
2	*	2				
3	*	2	*			
4	*	2	*	4		
5	*	2	*	4	*	
6	*	2	*	4	*	6

N = 6

odd j → star

even j → print j

N = 4 output

```
* ↴
* 2 ↴
* 2 * ↴
* 2 * 4 ↴
```

```
for (int i=1; i<=N; i++) {
    for (int j=1; j<=i; j++) {
        if (j%2 == 1) {
            sop("*");
        }
        else {
            sop(j);
        }
    }
    sopln();
}
```

i	j	
1	1	
	2	break
2	1	
	2	
	3	break
3	1	
	2	
	3	
	4	break

4	1 2 3 4 5 break
5	

② . Given N, print :

N = 3

```
*
* 1
* 1 *
```

N = 5

```
*
* 1
* 1 *
* 1 * 2
* 1 * 2 *
```

N = 6

```
*
* 1
* 1 *
* 1 * 2
* 1 * 2 *
* 1 * 2 * 3
```

for even j'

$\frac{j}{2}$ is correct

	1	2	3	4	5	6
1	*					
2	*	1				
3	*	1	*			
4	*	1	*	2		
5	*	1	*	2	*	
6	*	1	*	2	*	3

```
for (int i=1; i<= N; i++) {
        int k = 1;
        for (int j=1; j<=i ; j++) {
            if ( j%2 == 1) {
                Sop ("*");
            }
            else {
                Sop (k);
                k++;
            }
        }
        SOPln( );
}
```

N = 4 output

```
* ↴
* 1 ↴
* 1 * ↴
* 1 * 2 ↴
```

i	k	j
1	1	1
	1	2 break
2	1	1
	1	2
	2	3 break
3	1	1
	1	2
	2	3
	2	4 break
4	1	1
	1	2
	2	3
	2	4
	3	5 break

Q.3 Given N, print

N = 3

```
1
2  3
4  5  6
```

N = 4

```
1
2  3
4  5  6
7  8  9  10
```

N = 5

```
1
2  3
4  5  6
7  8  9  10
11 12 13 14 15
```

```
int k = 1;
for (int i = 1; i <= N; i++) {
    for (int j = 1; j <= i; j++) {
        SOP(k);
        k++;
    }
    SOPln();
}
```

N = 3

```
1
2 3
4 5 6
```

k = 1 2 3 4 5 6 7

i	j	
1	1	
	2	break
2	1	
	2	
	3	break
3	1	
	2	
	3	
	4	break

Q.4 Given N, print :

N = 3

```
_ _ *
_ * *
* * *
```

N = 4

```
_ _ _ *
_ _ * *
_ * * *
* * * *
```

N = 5

```
_ _ _ _ *
_ _ _ * *
_ _ * * *
_ * * * .
* * * * *
```

```
_ _ _ _ *
_ _ _ * *
_ _ * * *
_ * * * *
* * * * *
```

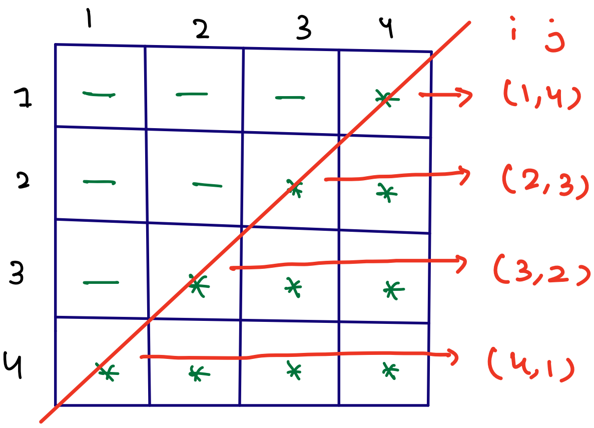
i	spaces	stars
1	4 → 5-1	1
2	3 → 5-2	2
3	2 → 5-3	3
4	1 → 5-4	4
5	0 → 5-5	5
	(N - i)	(i)

Each row = some spaces + some stars
(N - i) (i)

N = 3

```
for (int j=1; i<= N; i++) {
      // spaces
      for (int j=1; j<= N-i ; j++) {
            SOP(" ");
      }
      // stars
      for (int j=1; j<=i ; j++) {
            SOP("*");
      }
      SOPln();
}
```

if - else



	1	2	3	4
1	—	—	—	*
2	—	—	*	*
3	—	*	*	*
4	*	*	*	*

i j	i+j = N+1
(1,4)	1+4 = 5
(2,3)	2+3 = 5
(3,2)	3+2 = 5
(4,1)	4+1 = 5

```
for (int i = 1; i<= N; i++) {
    for (int j=1; j<= N; j++) {
        if ( i+j < N+1) {
            sop (" ");
        }
        else if ( i+j ≥ N+1) {
            sop ("*");
        }
    }
    sopln();
}
```

Q.5 Given N, print :

N = 3

```
*  *  *
—  *  *
—  —  *
```

N = 5

```
*  *  *  *  *
—  *  *  *  *
—  —  *  *  *
—  —  —  *  *
—  —  —  —  *
```

```
*  *  *  *  *
—  *  *  *  *
—  —  *  *  *
—  —  —  *  *
—  —  —  —  *
```

i	spaces		stars	
1	0	1-1	5	5-0
2	1	2-1	4	5-1
3	2	3-1	3	5-2
4	3	4-1	2	5-3
5	4	5-1	1	5-4
		(i-1)		N-(i-1) = N-i+1

each row = some spaces + some stars

(i-1) (N-i+1)

```
for (int i = 1; i <= N; i++) {
        // spaces
        for (int j = 1; j <= i-1 ; j++) {
                SOP (" ");
        }
        // stars
        for (int j=1; j <= (N-i+1) ; j++) {
                SOP ("*");
        }
        SOPln();
}
```

Q. 6 Given N, print :

N = 3

```
* * * * * *
* * _ _ * *
* _ _ _ _ *
```

N = 5

```
* * * * * * * * * *
* * * * _ _ * * * *
* * * _ _ _ _ * * *
* * _ _ _ _ _ _ * *
* _ _ _ _ _ _ _ _ *
```

N = 5

```
* * * * * | * * * * *
* * * * _ | _ * * * *
* * * _ _ | _ _ * * *
* * _ _ _ | _ _ _ * *
* _ _ _ _ | _ _ _ _ *
```

	1st part		2nd part	
i	st	sp	sp	st
1	5	0	0	5
2	4	1	1	4
3	3	2	2	3
4	2	3	3	2
5	1	4	4	1
	(n-i+1)	(i-1)	(i-1)	(n-i+1)

$$\underbrace{(i-1)\ (i-1)}_{2 * (i-1)}$$

```
for (int i=1; i<=N; i++) {
    // stars (left side)
    for (int j=1; j<=(N-i+1) ; j++) {
        sop("*");
    }
    // spaces
    for (int j=1; j<= 2*(i-1) ; j++) {
        sop("  ");
    }
    // stars(right side)
    for (int j=1; j<=(N-i+1); j++) {
        sop("*");
    }
    sopln();
}
```

Q. 7. Given N, print :

N = 5

```
*  -  -  -  -  |  -  -  -  -  *
*  *  -  -  -  |  -  -  -  *  *
*  *  *  -  -  |  -  -  *  *  *
*  *  *  *  -  |  -  *  *  *  *
*  *  *  *  *  |  *  *  *  *  *
```

	1st part		2nd part	
i	st	sp	sp	st
1	1	4	4	1
2	2	3	3	2
3	3	2	2	3
4	4	1	1	4
5	5	0	0	5
	i	N-i	N-i	i

2 * (N-i)

Q. 8 Diamond : join the above two logics