

GOOD EVENING

9:05 start

```
for(int i=1; i ≤ 10; i=i+2) {  
    SOP(i)  
}
```

$i: 1, 3, 5, 7, 9, 11 \Rightarrow \text{Break}$

// Q1: Given N , print $*$ N times

$N=5$, $*****$
 $N=9$, $*****$

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

- | | |
|-------------------------------------|-----------------|
| 1) Range $[1, N] \Rightarrow N$ | $i=1; i \leq N$ |
| 2) Range $[1, N-1] \Rightarrow N-1$ | $i=1; i < N$ |
| 3) Range $[0, N-1] \Rightarrow N$ | $i=0; i < N$ |

// Q2: Given N , print a square of $*$
of size $N * N$

$N=3$ $***$
 $***$
 $***$

$N=5$ $*****$
 $*****$
 $*****$
 $*****$
 $*****$

* * *

* * * * *

* * * * *

* * * * *

for(int j=1; j ≤ N; j++) {

// Print row of N *

for(int i=1; i ≤ N; i++) {

SOP(*)

}

Nested

SOPln()

loop

}

N = 3

j i

1 1

1 2

1 3

1 4 → Break

2 1

2 2

2 3

2 4 → Break

3 1

3 2

3 3

3 4 → Break

4 → Break

Output

* * * ←

* * * ←

* * * ←

// Q3: Given N, M
Print a rectangle of size $N \times M$

// $N = 2$ $M = 3$

```
* * *
* * *
```

// $N = 4$ $M = 2$

```
* *
* *
* *
* *
* *
```

$N \Rightarrow$ Rows

$M \Rightarrow$ Columns

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

```
    for (int col = 1; col <= M; col++) {
```

```
        SOP(*)
```

```
    }
```

```
    SoplN()
```

```
}
```

// Q4: Given N in input print the following

$N = 3$

```
*
* *
* * *
```

$N = 5$

```
*
* *
* * *
* * * *
* * * * *
```

Staircase Pattern

row	col	[N=3]
1	[1, 1]	
2	[1, 2]	
3	[1, 3]	

```
for (int row=1; row ≤ n; row++) {
```

```
    for (int col=1; col ≤ row; col++) {
```

```
        SOP( * )
```

```
    }
```

```
    SOPLn()
```

```
}
```

row

col

Output

1

1

* ←

1

2 → Break

* * ←

2

1

* * * ←

2

2

2

3 → Break

3

1

3

2

3

3

3

4 → Break

4

→ Break

// Q5: Given N in input print the following

$N=3$

```

* * *
* *
*

```

$N=5$

```

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

```

[$N=3$]

row	col	$n+1-\text{row}$	$\text{row} + \text{col} = n+1$
1	3	$3+1-1 = 3$	$\text{col} = n+1 - \text{row}$
2	2	$3+1-2 = 2$	
3	1	$3+1-3 = 1$	

```

for( int row=1; row ≤ n; row++ ) {
    for( int col=1; col ≤ n+1-row; col++ ) {
        SOP( * )
    }
    SOP( \n )
}

```

Break : 10:40

// Q6: Given N in input print the following

$N = 4$

```

*
* 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()
```

```
}
```

Pattern with spaces

11 Q7: $N = 3$

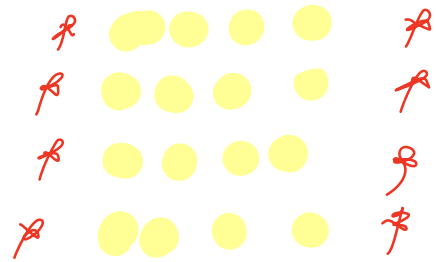
```

*   *
*   *
*   *
  
```

$N = 5$

```

*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
  
```



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

```
    SOP(*)
```

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

```
        SOP( )
```

```
    }
```

```
    SOP(*)
```

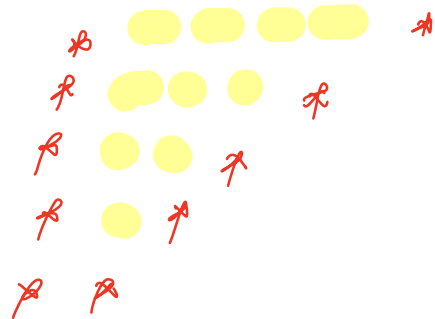
```
    SOP(n)
```

```
}
```

// Q8: $N=3$



$N=5$



[$N=5$]

row

spaces

1
2
3

4
3
2

row + space = n

space = n - row

4
5

1
0

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

```
    SOP(*)
```

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

```
        SOP( )
```

```
    }
```

```
    SOP(*)
```

```
    SOPLnL)
```

```
}
```

// Q9: $N = 3$

```

  ● ● *
  ● * *
  * * *
```

$N = 5$

```

  ● ● ● ● *
  ● ● ● * *
  ● ● * * *
  ● * * * *
```

[$N = 5$]

Row	Spaces	Stars
1	4	1
2	3	2
3	2	3
4	1	4

5

0

5

 $n - \text{row}$ row

$$\text{row} + \text{spaces} = n$$

$$\text{spaces} = n - \text{row}$$

```
for (int row = 1; row ≤ n; row++) {
```

```
    // Print space
```

```
    for (int col = 1; col ≤ n - row; col++) {
```

```
        SOP(" ")
```

```
    }
```

```
    // Print star
```

```
    for (int col = 1; col ≤ row; col++) {
```

```
        SOP("*")
```

```
    }
```

```
    SOP("\n")
```

```
}
```

T testcases : <https://www.interviewbit.com/snippet/240a968a9752a6936775/>

Pattern : <https://www.interviewbit.com/snippet/6201c8dbc56c9c317fb5/>