

row \ col	1	2	3	4	5
1	11	12	13	14	15
2	21	22	23	24	25
3	31	32	33	34	35
4	41	42	43	44	45
5	51	52	53	54	55

→
if (col >= row)
else " "

if (row >= col)
printf("#");

#

#

if (i <= (n+1))

#

if (i >= (n+1))

#

else " "

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

1
2 2
3 3 3
4 4 4 4
5 5 5 5 5

(2 * row - 1) ⇒ 4

1-3

1
2 3
4 5 6
7 8 9 10

i=2
j=2-3
j=3-3

```
for (int i = 1; i <= n - 1; i++)
{
    // print spaces before first star
    for (int j = 1; j <= i; j++)
    {
        printf(" ");
    }
    for (j = 1; j <= 2 * i - 1; j++)
    {
        printf("#");
    }
}
```


#

	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	1	2	3	4	5	6	7
3	1	2	3	4	5	6	7
4	1	2	3	4	5	6	7

```
int main()
{
    int n;
    printf("Enter n: ");
    scanf("%d", &n);
    int row = 1;
    while (row <= n)
    {
        int col = 1;
        while (col <= 2 * row - 1)
        {
            printf("%d ", col);
            col++;
        }
        printf("\n");
        row++;
    }
    return 0;
}
```

4 3 2 1

① (n-1) + (n-2) + (n-3) + ... + 2 + 1

1 + n = $\frac{n(n+1)}{2}$

$O(n^2)$ = $\frac{n(n+1)}{2}$

Solve (n) + (n-1) + (n-2) + ... + 1 = $O(n^2)$

row \ col	1	2	3	4
1	1	2	3	4
2	1	2	3	4
3	1	2	3	4
4	1	2	3	4

```
int main()
{
    int m, n;
    printf("Enter m and n: ");
    scanf("%d %d", &m, &n);
    int c[m][n];
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            c[i][j] = m[i][j] * n[j][i];
        }
    }
}
```