this is for n= 9

for in↗ nos. → *
  dec ___ ___ #

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


Identify tasks.

→ no. of lines = 5 , no. of cols = 9 = (2×5−1)

↓ rows-2×rowno.↑
9 —

rowno.	Star	space	hashes
1	1	7	1
2	2	5	2
3	3	3	3
4	4	1	4
→5	5	0	4

stars = rowno. ;

hashes = rowno.
{ if rowno = no. of lines
  (hashes = rowno. −1) }

cols − (stars +hashes)
→

{ spaces = (-1) +7 → + exactly X.
  while (spaces > 0) }

for each row → Stars , spaces, hash.

n = 9;

* * * * * * *
            *

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

empty /
hollow hourglass

if (first || last)
  cout << *
else
  cout << " "

1 2 3 4 5 6 7.

cols = 7

2×lines −7
= ⑦

for uppercase   2 tasks → spaces, stars.
                 no of lines → 4

row no.	spaces	stars.
1	0	7
2	1	5
3	2	3
4	3	1

Ⓧ spaces 'in'

form for spaces → row no. −1;
        stars → ⑦ − 2(row no. −1)
                2×lines −1 − 2× row + 2
              → 2(lines−row) +1
                2× 4 − 7

row no.

7.

rowno.

1

2   7

              5

2×lines – and.rowno. → col of last star.

we will find the form for 1st &
last star that we want to print.

           1   2   3   4   5   6   7   tot. cols = 7

$2×4-2$ 1 * * * * * * *   lines = 4

= 6th col. 1   rowno = 2

                  *   *   2×lines – rowno

       2.   *   * $2×4=8-3=$ ⑤

$2×4-4$ 3-   *   "if row = 1

= ④   4

            2,2 , 3,3 , 4,4

first star col. no is =) <u>rowno</u> .

last star →   row = 2 → col. 6.

                 row = 3 → coln 5

                 row = 4 → coln 4.

| rowno, lines, total columns. | → generalised form

last star → 2×lines – rowno .