

Try following patterns:

1.

```
input.txt  X
input.txt
1  5

output.txt  X
output.txt
1      *
2     * *
3    * * *
4   * * * *
5  * * * * *
6
```

2.

```
input.txt  X
input.txt
1  5

output.txt  X
output.txt
1      1
2     1 2
3    1 2 3
4   1 2 3 4
5  1 2 3 4 5
6
```

3.

```
input.txt  X
input.txt
1  5

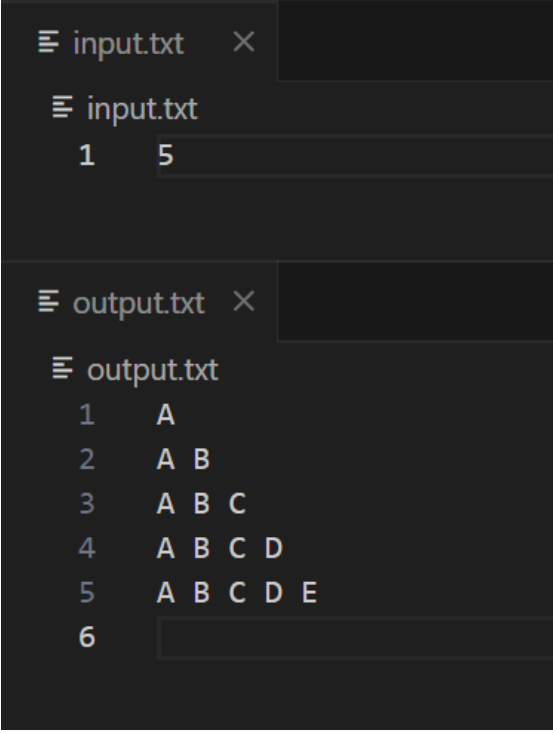
output.txt  X
output.txt
1  *****
2  | *****
3  |   ****
4  |   ***
5  |   **
6  |   *
   |
```

4.

```
input.txt  X
input.txt
1  5

output.txt  X
output.txt
1  *****
2  | *****
3  |   ****
4  |   ***
5  |   **
6  |   *
   |
```

5.



The image shows a code editor with two tabs: 'input.txt' and 'output.txt'. The 'input.txt' tab is active and contains the number '5'. The 'output.txt' tab is also visible and contains a sequence of numbers 1 through 6, each followed by a space and a string of letters. The strings are 'A', 'A B', 'A B C', 'A B C D', and 'A B C D E' for lines 1 through 5 respectively. Line 6 is currently empty.

```
input.txt ×
input.txt
1 5

output.txt ×
output.txt
1 A
2 A B
3 A B C
4 A B C D
5 A B C D E
6
```