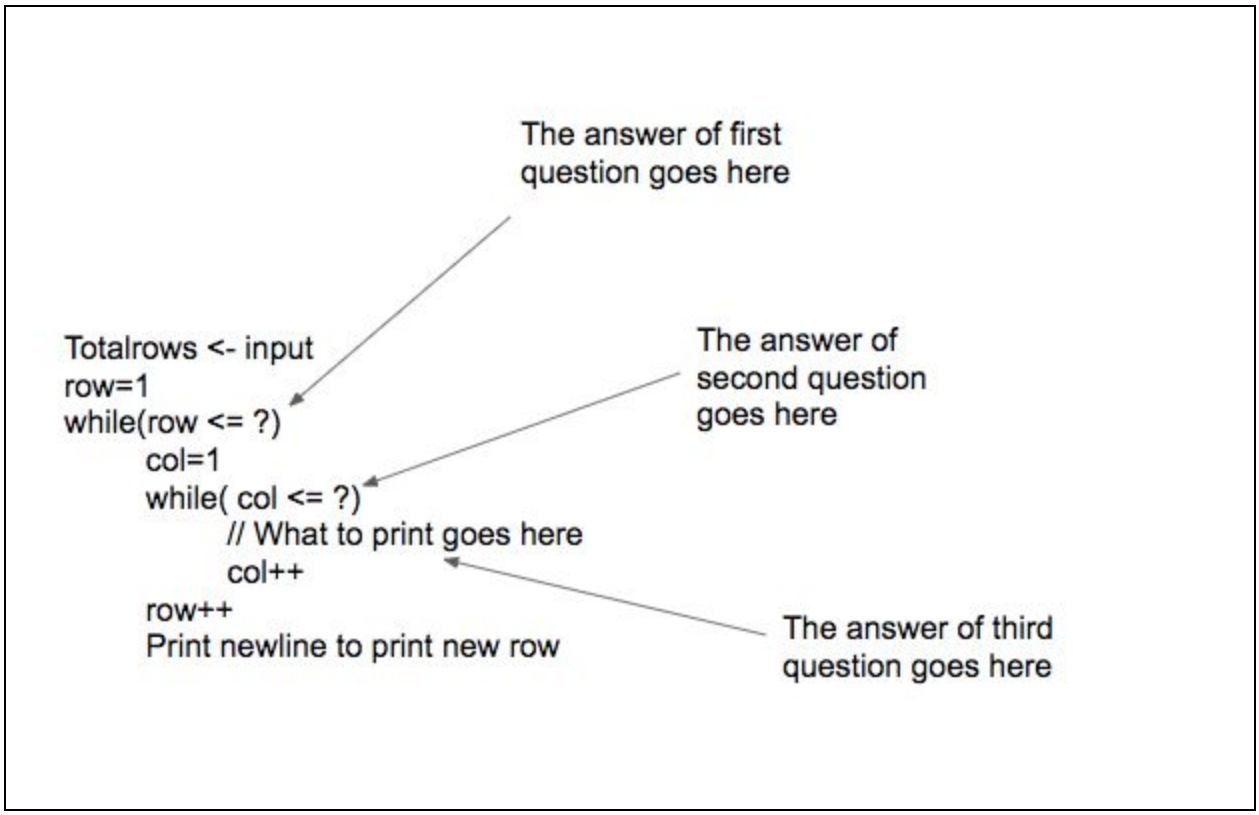
Pattern: Triangle of Numbers

Problem Description: You are given with an input number N, then you have to print the given star pattern corresponding to that number N.
For example, if N=3
Pattern output:
```
  1
 232
34543
```

For N=4, the pattern output would be:
```
   1
  232
 34543
4567654
```

Generic approach to solve pattern questions:

1. For solving pattern questions, you have to answer three questions. The first question is how many rows are to be printed in the given pattern.
2. The second question is how many columns are to be printed in a generic row of the given pattern.
3. The third question is what to print at generic row and generic column location.
4. The answer to these questions form the basis of implementing a pattern.
5. The generic structure of code, after answering these three questions, looks like this:



Answers to these three questions for the given pattern are:

1. The number of rows to be printed are given as input in N.
2. The number of columns to be printed in a generic row have to be analysed through two examples:
   Let us see the pattern for N=3 (in the following figure, underscore represents space)

__1

_232

34543

Row Number	Number of columns printed
1	3 (2 spaces and 1 number)
2	4 (1 space and 3 numbers)
3	5 (5 numbers)

From the tabular data, it can written that for a generic row r, number of columns to be printed are N + r - 1.

Let us see if this holds true for N=4

N=4

_ _ _1

_ _ 2 3 2

_ 3 4 5 4 3

4 5 6 7 6 5 4

Row Number	Number of columns printed	Number of columns printed according to expression (N+r-1)
1	4 (3 spaces and 1 number)	4
2	5 (2 spaces and 3 numbers)	5
3	6 (1 space and 5 numbers)	6
4	7 (7 numbers)	7

This proves that given expression holds true. So, for a generic row, number of columns to be printed are N + r - 1.

3. In this pattern, we have to print three things: spaces, increasing numbers and decreasing numbers.
   For any generic row r and generic column c,
   a. space is printed on columns less than or equal to N - r
   b. Increasing numbers start from r and is written on columns [N - r + 1, N]
   c. Decreasing numbers are printed for remaining columns.

Pseudo code for the given problem:

```
input=N
i=1
while i is less than or equal to N:
        j=1
        num=i
        while j is less than or equal to N + r - 1:
            If (j <= (N-i))
                Print " "
```

```
        Else if(j < N)
            Print num
            num=num+1
        Else
            Print num
            num=num-1
        Increment j by 1
    Increment i by 1
    Add a new line here
```

Approach 2:

How to approach?

1. Take N as input from the user.
2. Figure out the number of rows, (which is N here) and run a loop for that.
3. Now, figure out how many columns are there in ith row. Here, we need to print in 3 different loops. One loop is for spaces, another 2 loops are for printing numbers, first in increasing order and then in decreasing order.
4. Now, figure out "What to print?" in a particular (row, column). Here, we have to print N-i spaces first, then numbers in increasing order upto i columns, such that the first number in a row is equal to the row number, then we start decreasing the numbers till i-1 columns again.

Pseudo code for the given problem:

```
input=N
i=1
While i is less than equal to N:
    spaces=1
    While spaces is less than (n-i):
        print(' ')
        Increment spaces by 1
    temp=i
    j=1
    While j is less than equal to i:
        print(temp)
        Increment temp by 1
    Decrement temp by 1
    k=1
    While k is less than i:
        Decrement temp by 1
```

print(temp)

Increment i by 1

Add a new line here

❏ Let us dry run the Code for N=4

- i=1(<=4)
  - ➔ 4-1=3 spaces are getting printed first.
  - ➔ j=1(<=1), so print 1
  - ➔ j=2 (>1), move out of the inner loop with a new line
  - ➔ k=1(>=1), move out of the inner loop with a new line

- i=2(<=4)
  - ➔ 4-2=2 spaces are getting printed first.
  - ➔ j=1 (<=2), so print 2
  - ➔ j=2 (<=2), so print 3
  - ➔ j=3 (>2), move out of the inner loop with a new line
  - ➔ k=1(<2), so print 2
  - ➔ k=2(>=2), move out of the inner loop with a new line

- i=3(<=4)
  - ➔ 4-3=1 space is getting printed first.
  - ➔ j=1(<=3), so print 3
  - ➔ j=2(<=3), so print 4
  - ➔ j=3(<=3), so print 5
  - ➔ j=4(>3), move out of the inner loop with a new line
  - ➔ k=1(<3), so print 4
  - ➔ k=2(<3), so print 3
  - ➔ k=3(>=3), move out of the inner loop with a new line

- i=4(<=4)
  - ➔ 4-4=0 no space is getting printed.
  - ➔ j=1(<=4), so print 4
  - ➔ j=2(<=4), so print 5
  - ➔ j=3(<=4), so print 6
  - ➔ j=4(<=4), so print 7
  - ➔ j=5(>4), move out of the inner loop with a new line
  - ➔ k=1(<4), so print 6
  - ➔ k=2(<4), so print 5
  - ➔ k=3(<4), so print 4
  - ➔ k=4(>=4), move out of the inner loop with a new line

- i=5(>4), move out of the loop

So , final output:

```
    1
  232
 34543
4567654
```