```
    Make a guess (1 to 10): 3
    Wrong guess; current credit is 2.00!
    Make a guess (1 to 10): 10
    Wrong guess; current credit is 1.50!
    Make a guess (1 to 10): 1
    Wrong guess; current credit is 1.00!
    Make a guess (1 to 10): 2
    Wrong guess; current credit is 0.50!
    Make a guess (1 to 10): 8
    Wrong guess; current credit is 0.00!
    No more credits!
    You have played 9 rounds!
```

💻 Exercise 7.2.3 ***

Write a Python function `print_tree` that repeatedly prompts the user for a positive integer *n*.

The function then generates a tree with *n* rows, with the 1st row having a single star, the 2nd row having two stars, the 3rd row having 3 stars, and so on…, and the *n*th row having *n* stars.

If there are multiple stars in a row, then each of the stars is separated by a space between them.

The function stops asking the user for more numbers, if the user types 'No' when prompted for more numbers.

You should use only **while** loops (and not for loops) for this question.

The following shows an example of a tree with n=6.

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

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

Following is helpful to find the laws in the printing results.

```
        *   5s 1*
       * *  4s 1*6*
      * * *
     * * * *
    * * * * *
   * * * * * *
```

(add-on: think about how to make a hollow tree based on your code...)

Here is some code to get you started:

```python
def print_trees():
    # write your code here
    return
```

Here is an example of how the output may look like.

```
Input a positive integer, or type No to quit: 6
     *
    * *
   * * *
  * * * *
 * * * * *
* * * * * *


Input a positive integer, or type No to quit: 7
     *
    * *
   * * *
  * * * *
 * * * * *
* * * * * *
* * * * * * *
Input a positive integer, or type No to quit: No
```