UCF "Practice" Local Contest — Aug 23, 2014

Sierpiński Triangle

filename: `triangle`
(Difficulty Level: `Medium`)

The Sierpiński triangle is a beautiful fractal found in mathematics. As with many fractal patterns, it is constructed by starting with a given shape, applying a function to that shape, then applying the same function to the resulting shapes, and so on. In theory, this function is applied infinitely many times, but in practice it usually stops after a given number of applications since the resulting shapes get too small to be noticeable.

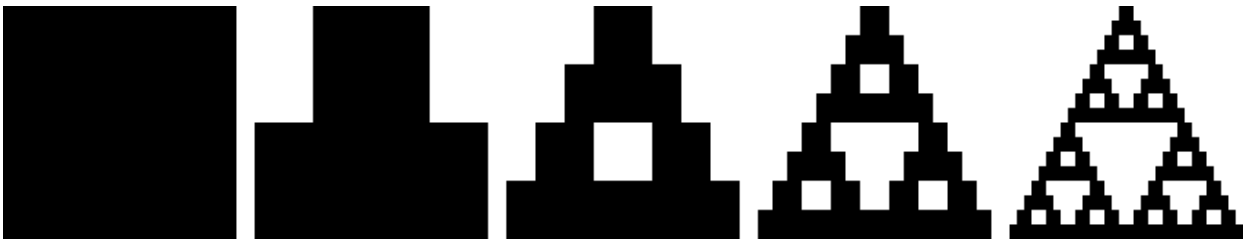
The Sierpiński triangle is created with the following steps:

1) Start with an equilateral triangle with the base parallel to the horizontal axis.
2) Create an upside down triangle with half the height and width of the original triangle and cut this pattern out of the center of the original triangle, leaving 3 equilateral triangles.
3) Repeat step 2 on each of the newly formed triangles.

The following image (from Wikipedia) shows triangles of level 1 through 5, respectively:

Sierpiński Triangle

Interestingly enough, there are other shapes to which you can apply a similar pattern that result in close approximations of a Sierpiński triangle. For example, starting with a square, you can remove rectangles of half the height and a quarter of the width from the upper left and right corners, resulting in three new squares. The following image (from Wikipedia) shows approximate triangles of level 1 through 5, respectively:



Approximate Sierpiński Triangle

The Problem:

Given a level, draw an ASCII version of the approximate Sierpiński triangle.

The Input:

The input will begin with a positive integer T representing the number of triangles to draw. This will be followed by T lines each with a positive integer $K \leq 10$ representing the level of the triangle you are to draw.

The Output:

For each triangle, begin with the line "`Triangle #`*S*`:`" with S starting at 1. Then output the ASCII version of the approximate level K triangle, using the character 'X' to represent the drawn pattern and spaces elsewhere. You should scale the image so that the smallest square regions are 2x2. The bottom left point of the image should be in column 1. See the sample output for clarification. Follow each triangle with a blank line.

Sample Input:

```
2
1
4
```

Sample Output:

```
Triangle #1:
XX
XX

Triangle #2:
       XX
       XX
      XXXX
      XXXX
     XX  XX
     XX  XX
    XXXXXXXX
    XXXXXXXX
   XX      XX
   XX      XX
  XXXX    XXXX
  XXXX    XXXX
 XX  XX  XX  XX
 XX  XX  XX  XX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
```

```
Triangle #1:
XX
XX

Triangle #2:
.......XX.......
.......XX.......
......XXXX......
......XXXX......
.....XX..XX.....
.....XX..XX.....
....XXXXXXXX....
....XXXXXXXX....
...XX......XX...
...XX......XX...
..XXXX....XXXX..
..XXXX....XXXX..
.XX..XX..XX..XX.
.XX..XX..XX..XX.
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
```

Note: The first column shows what you should actually output. The second column shows a '.' where all spaces occur for clarification. Do not submit a solution with periods for spaces – you will receive wrong answer for this.