

Problem 1

Pascal's triangle contains the binomial coefficients:

$$\begin{array}{rcccccccc}
 r = 0 & & & & & & & 1 \\
 r = 1 & & & & & 1 & 1 & \\
 r = 2 & & & 1 & 2 & 1 & & \\
 r = 3 & & 1 & 3 & 3 & 1 & & \\
 r = 4 & & 1 & 4 & 6 & 4 & 1 & \\
 r = 5 & & 1 & 5 & 10 & 10 & 5 & 1 \\
 r = 6 & 1 & 6 & 15 & 20 & 15 & 6 & 1 \\
 r = 7 & 1 & 7 & 21 & 35 & 35 & 21 & 7 & 1
 \end{array}$$

where

$$p_{r,c} = \binom{r}{c} = \frac{r!}{c!(r-c)!}. \quad (1)$$

The coefficients can easily be computed from the recurrence relation

$$p_{r,c} = \begin{cases} 1, & \text{if } c = 1 \text{ or } c = r \\ p_{r-1,c-1} + p_{r-1,c}, & \text{otherwise.} \end{cases} \quad (2)$$

Write a class `pascal` so that `pascal(n)` is the object containing `n` rows of the above coefficients. Include the following methods:

1. `get(i,j)`, which returns the coefficient at position `(i,j)` (row,column).
2. `printFull`, to print the full display of Pascal's Triangle (shown below, left).
3. `print(int m)`, which prints a star if the coefficient modulo `m` is nonzero, and a `space` character otherwise. As an example, the output of the method with `m = 2` would be (shown on right):

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1

```

Example of `printFull`

```

      *
     * *
    * *
   * * *
  * * * *
 * * * * *
* * * * *

```

Example of `print(2)`

Note: The object needs to have an array internally. The easiest solution: make an array of size $n \times n$.