

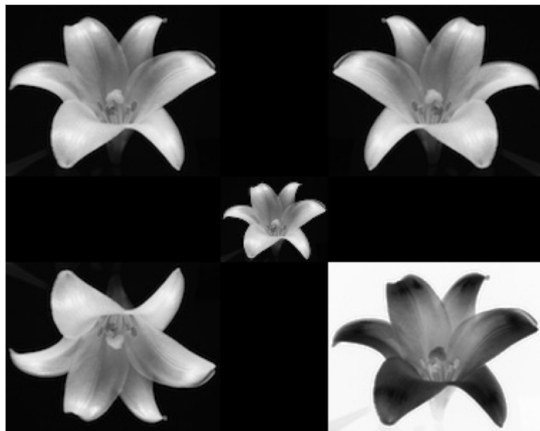
(1) Create a composite picture

Suppose the variable `flower` is assigned to a 120x150 matrix storing an image of a flower:

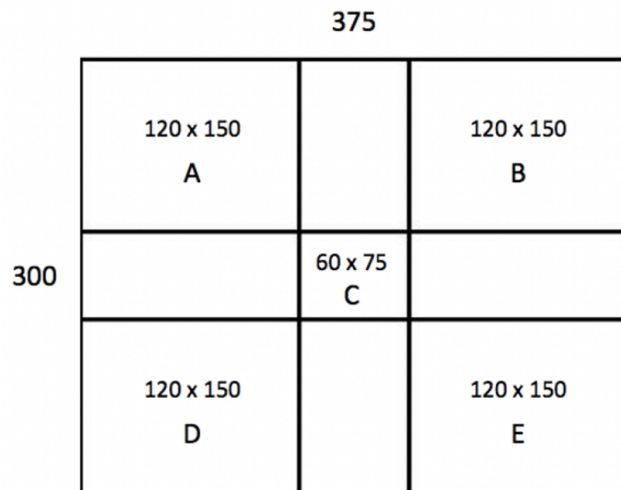


`flower`

We'll work together to write a sequence of statements to create the 300 x 375 image below, named `garden`, which is a composite of flower images. The flowers in the four corners of the garden image are the same size as the original image (120x150), as shown in the diagram on the right. The flower in region A is a copy of the original, the flower in region D (lower left) is upside down, the flower in region B is mirror-reversed in the horizontal direction, the flower in region E is a *negative* of the original image, and the flower in the center is half the size.



`garden`



(2) Define a function named `d2g` that has two inputs, `sigma` and `x`, and returns the value of the second derivative of a Gaussian function, given by the following expression:

$$\left(\frac{x^2}{\sigma^2} - 1\right)e^{-\frac{x^2}{2\sigma^2}}$$

(3) Write a test script for `d2g` that creates a 21-element vector of `d2g` values for `x` ranging from -10 to +10 (use a `for` loop for this)

(4) Define a function `avgImage` that has two inputs, an image and a neighborhood size, and returns a new image where each location stores an average of the values in the neighborhood.