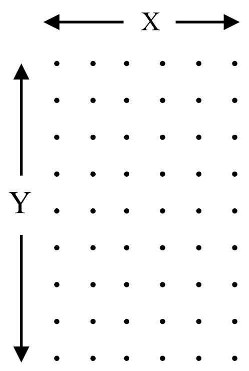
Two Dimensional Arrays (Matrices) (Outsource: 8-32 - 8-38)

A matrix, or two-dimensional array, is simply an array of arrays. It contains rows and columns, with the rows (y) going up and down, and the columns (x) going across.

Here is an example of a matrix that has 9 rows and 6 columns. It is a 9 X 6 matrix that is filled with dots.



To declare this matrix in Java, you must use the **new** keyword:

```
char grid[][] = new char[9][6];
```

The **char** indicates what data type the matrix will contain. The two numbers in braces indicate how many ROWS and COLUMNS there are to be in the matrix.

To fill it with dots you would use a nested loop:

```
for (int y = 0; y < grid.length; y++)           // Controls the number of rows (y)
    for (int x = 0; x < grid[y].length; x++)    // Controls the number of columns (x)
        grid[y][x] = '.';                       // Assigns a . to the specified element
```

To output the contents of this matrix, use a nested loop:

```
for (int y = 0; y < grid.length; y++)
{
    for (int x = 0; x < grid[y].length; x++)
        System.out.print( grid[y][x] );
    System.out.println();                       // Inserts a new line after each row
}
```

Notice that when you refer to each matrix element, you must use two brackets. The first one indicates the ROW, and the second one the COLUMN.

To change the contents of the grid, you can access locations individually.

To assign the letter 'A' to the lower right corner of the grid, use the following command:

```
grid[8][5] = 'A';
```

Now the matrix looks like this:

```
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . A
```

Here is how you would fill row 1 with the letter 'A':

```
for (int x = 0; x < grid.length; x++)
    grid[1][x] = 'A';
```

The resulting matrix is:

```
. . . . . .
A A A A A A
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . A
```

Now let's fill the last COLUMN with the letter Z.

```
for (int y = 0; y < grid.length; y++)
    grid[y][5] = 'Z';
```

3
The resulting matrix should look like this:

```
A . . . . Z
A A A A A Z
. . . . . Z
. . . . . Z
. . . . . Z
. . . . . Z
. . . . . Z
. . . . . Z
. . . . . Z
```

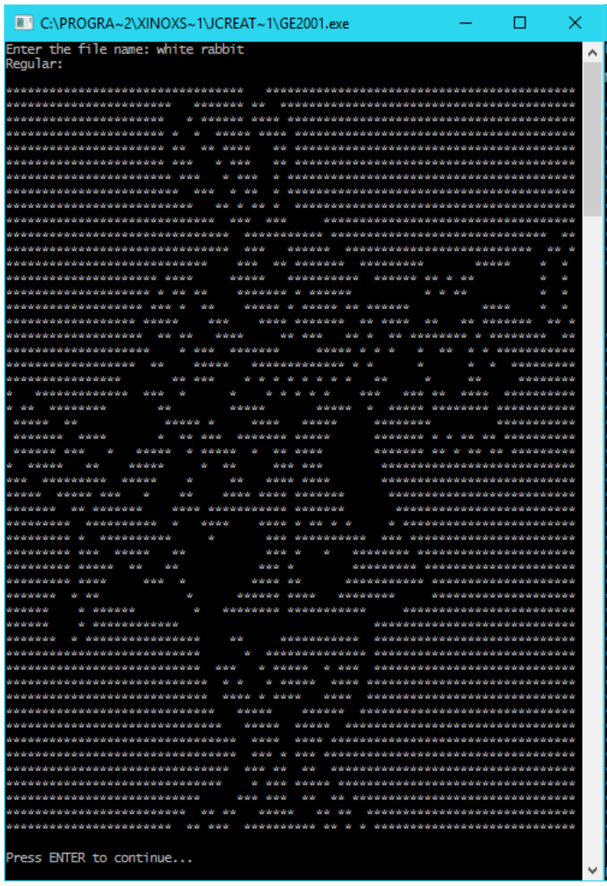
Matrix Rotation

Matrix Rotation

OBJECTIVE

Complete the following methods in the **MatrixRotation** class:

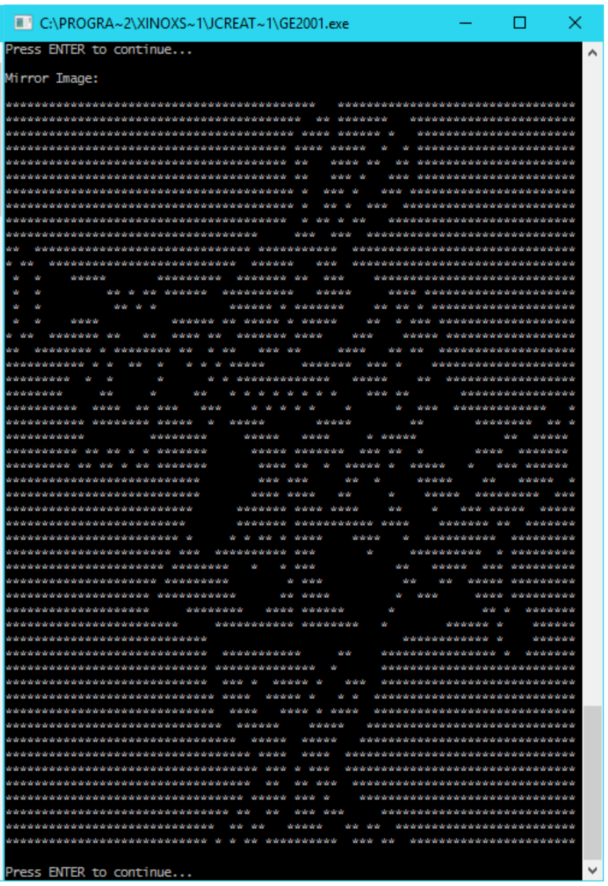
boolean input()	// opens a text file and fills a 2D array with spaces and #'s.
void regular()	// displays the matrix (as is).
void upsideDown()	// displays the matrix upside down
void right90()	// rotates the image right 90º
void left90()	// rotates the image left 90º
void mirror()	// displays a mirror image
void doubleInverted()	// displays the image upside down and mirrored



```
Enter the file name: white rabbit
Regular:

Press ENTER to continue...
```

Regular Image



```
Press ENTER to continue...

Mirror Image:

Press ENTER to continue...
```

Mirror Image