Analysis of Algorithms

AA Lab 1a

1 Matrices

In order to create a two dimensional array, we use:

```
int[][] myMatrix = new int[<width>][<height>]
```

For example, to create an array with width 4 and height 6, we would use:

```
int[][] myMatrix = new int[4][6]
```

Then, in order to refer to the items in the matrix, we can use:

```
int[][] myMatrix = new int[<x>][<y>]
```

For example, if we wanted to access the item in column 2, row 5, we would use:

```
int i = myMatrix[2][5]; //reads an item from the matrix
myMatrix[2][5] = 7; //writes an item to the matrix
```

2 Lab 2a

2.1 Submission: Snake

The game of snake has been around a long time. The objective of the game is to guide your snake around the play area, avoiding obstacles such as your own tail and walls, while at the same time eating the apples which help you to grow longer.

Your task is to take in some information which will help you construct a simple visualization of the snake game.

2.1.1 Input

You will be given a few lines of input. The first line contains 4 integer values - the number of snakes, the width and height of the playing area, and a fourth number which is always 1. This line will be followed by as many lines as there are snakes.

For each snake, there will be a line containing information about the snake. It consists of a series of x,y pairs separated by spaces, which describe important points of the snake. The first x,y point will be the snake's head position. The last x,y point will be the tail of the snake. All points between the head and the tail represent bends in the snake.

2.1.2 Output

Write a class called `Program.java` that will print out a matrix that represents the snake game. All points with no snake in it must be 0's and all points with a snake in it must contain the snake number. Note that snakes are numbered from 1 rather than 0. All points must have a space between it and the next point. The top left of the matrix is at point 0,0 and the bottom right is at point width-1,height-1.

2.1.3 Sample Input

```
3 15 15 1
1,1 2,1 2,12 8,12 8,0 3,0
10,2 13,2 13,5
14,14 14,10 12,10
```

2.1.4 Sample Output

```
0 0 0 1 1 1 1 1 1 0 0 0 0 0 0
0 1 1 0 0 0 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0 1 0 2 2 2 2 0
0 0 1 0 0 0 0 0 1 0 0 0 0 2 0
0 0 1 0 0 0 0 0 1 0 0 0 0 2 0
0 0 1 0 0 0 0 0 1 0 0 0 0 2 0
0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0 1 0 0 0 3 3 3
0 0 1 0 0 0 0 0 1 0 0 0 0 0 3
0 0 1 1 1 1 1 1 1 0 0 0 0 0 3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 3
```

2.1.5 Structure

You should create a method called `drawSnake`, which takes in a String corresponding to one line of input representing a Snake, an integer representing the snake number, and a matrix representing the play area. This method should then draw the snake onto the play area. You can then call this method for each line of snake input in order to draw every snake onto the matrix.

To help with the `drawSnake` method, create a method called `drawLine`, which takes in the matrix representing the play area, two Strings and an integer. The two Strings represent points (For instance `10,2` and `13,2`) and the integer is the snake number. The method puts the snake number in every position between the two points (including the points themselves). In order to do this, find the minimum x value in these two points and the maximum x value in these two points. Do the same for y. Then set all array positions which are both between the minimum x value and maximum x value, and between the minimum y position and maximum y position to the snake number.

You should also create a method called `printBoard`, which takes in the matrix representing the board and prints it out in the required format.

2.1.6 Hint

The easiest way to do the `drawSnake` method is to split the line of input given to the method by the space character. This will result in an array in which each String represents a two dimensional point. Now, for every two consecutive points, call the `drawLine` method.

One task you will need to accomplish is splitting the string so that you can identify the individual snake segments. In the string above, you will need to seperate the `1,1` from the `2,1` and so on. The split function can be very useful for this. It takes as input a substring to split on. An example of how this can be used is shown below:

```
String test="43,19,27,32";
String[] out = test.split(",");
```

After this code executes, the array out will contain the substrings 43, 19, 27 and 32. So

```
System.out.println(out[2]);
```

would print out `27`.

You will need to split by the space character in the `drawSnake` method, and by the comma character in the `drawLine` method.