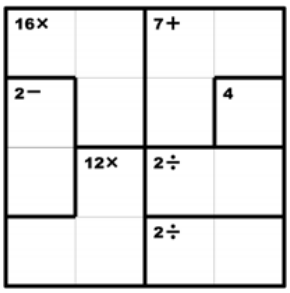
CSCI 340
Data Structures and Algorithms
Spring 2019
Project 5 – KenKen

Points: 50
Due date: Thursday, April 25

This assignment borrows from one used by Prof. David Reed at the University of Nebraska – Omaha.

For this assignment, you are to design and implement a Java program for solving KenKen puzzles. The class should be named `KenKen`. The file should be `KenKen.java`. If you are not familiar with KenKen, it is played on an NxN puzzle grid in which the numbers 1 through N are placed.



To complete a puzzle, the player must fill in the grid such that the numbers 1 through N appear in every row and column. Furthermore, sets of outlined squares (called cages) have mathematical constraints. For example, the top left square in the above puzzle is in a cage with the constraint "16x," which means that the three numbers in the cage have a product of 16 (i.e., 4*2*2 or 4*4*1).

For a person, solving a KenKen puzzle requires complex and careful logical reasoning. Try so solve a few puzzles on the web. For example, https://www.kenkenpuzzle.com/ posts multiple puzzles every day.

KenKen puzzles can be easily solved by a computer using depth first search with recursive backtracking.

Your program should prompt the user for a file that contains the specifications for a puzzle. The first line of the puzzle file should specify the size of the puzzle (you may assume a maximum size of 9x9). Each subsequent line should identify a cage, with the constraint first (e.g., "16*") followed by the coordinates in the cage. For example, the above puzzle would be represented as:

```
4
16 * 0 0 0 1 1 1
7 +  0 2 0 3 1 2
2 -  1 0 2 0
4 #  1 3
12 * 2 1 3 0 3 1
2 / 2 2 2 3
2 / 3 2 3 3
```

Notes:

- To make things easier to parse, the cells appear as pairs of numbers separated by whitespace. Thus, the line `16 * 0 0 0 1 1 1` states that the cells `(0,0) (0,1)` and `(1,1)` multiply to be `16`.
- '-' and '/' constraints always contain exactly two points
- We use the '#' to represent no-op. No-ops constraints always contain exactly one cell.

Your program should display the solved puzzle if a solution exists, or display a message if no solution is possible. The solution to the above puzzle is:

```
2 4 1 3
1 2 3 4
3 1 4 2
4 3 2 1
```

How to Submit:

Name your program `KenKen.java`. Do NOT put a package statement in your code. Submit your code on the lab machines. Do not submit an entire project, just your java file. E.g. At a linux prompt type:

```
submit 340 KenKen.java
```