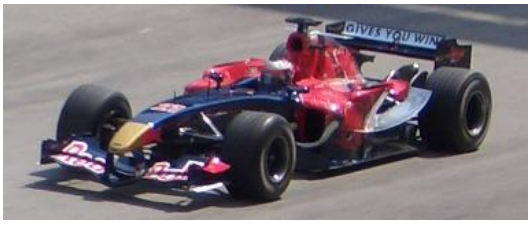
Problem 129: Code Racer

Difficulty: Hard

Author: Richard Green, Whiteley, Hampshire, United Kingdom

Originally Published: Code Quest Australia 2019

Problem Background



Video gaming is a huge industry, growing larger every day, particularly with the rise of professional "E-sport" leagues. Despite how complex and realistic video games have become, their beginnings were much humbler. Today, we'll be creating a retro-style video game much like those earliest games.

"Code Racer" requires the player to drive their car through a race track filled with obstacles. They win the game if they can successfully make it to the end of the track. If they hit any obstacles, however, it's game over.

Problem Description

Your program must print the player's journey through the race track and determine if the player won the game or crashed. The program's input will include information about the size of the race track, rules defining where obstacles are to be placed along the track, and a list of commands from the player to steer their car.

Obstacle rules are presented as a pair of integers, representing how often the rule applies, and in what column an obstacle should appear. For example, the obstacle rule "4 1" indicates that an obstacle should appear in every fourth line after the starting line in the first column.

The player's commands will consist of the uppercase letters "L" and "R", and spaces. These indicate whether the car should move one column to the left, one column to the right, or remain in the same column, respectively. Any command that would force the car outside the bounds of the race track should be ignored.

Your program will need to print out the starting line and as much of the race track that the player successfully navigated, showing the location of the player's race car at each point throughout the track. The format of the starting line consists of three lines of text:

- A line of equals signs (=), two more than the width of the track
- A line with a pipe (|), a number of spaces equal to the width of the track, except for a lowercase letter 'v' at the race car's starting location, followed by another pipe
- A line of dashes (-), two more than the width of the track

See the sample output below for an example.

When the player reaches the end of the race track, they have won the game. Your program should congratulate them by printing the following text to show the finish line:

- A line of equals signs (=), two more than the width of the track
- A line reading "Course Complete!"

If the player crashes into an obstacle, you should print the line in which the crash occurred as normal (according to the format described below), replacing the offending obstacle with a capital letter X to show the location of the crash, followed by a line that reads "You Crashed - GAME OVER". Any directional commands after that point in that test case should be ignored.

Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include the following lines of output:

- A line containing three positive integers, separated by spaces.
    - The first integer represents the width of the race track, W.
    - The second integer represents the length of the race track, L.
    - The third integer represents the race car's starting position, S. 1 represents the left-most column. This value will be between 1 and W inclusive.
- A line containing a single positive integer, R, representing the number of obstacle rules used by this track.
- A total of R lines containing obstacle rules for the track. Each obstacle rule consists of two positive integers, separated by spaces, on a single line.
    - The first integer represents how often this rule applies, expressed in a number of rows after the starting line. 1 represents every row, 2 represents every other row starting with the second, etc. This value will be between 1 and L inclusive.
    - The second integer indicates the column index in which an obstacle should appear on these lines. 1 represents the left-most column. This value will be between 1 and W inclusive.
- A line containing L characters representing player commands. These will consist of spaces and the uppercase letters "L" and "R". Warning: This line may end with one or more spaces, however the sample input below does not.

```
2
11 10 6
2
4 1
5 3
 LLLLLLRRR
```

```
11 10 7
2
4 1
5 3
 LLLLLLRRR
```

Sample Output

For each test case, your program must output a representation of the race track, including as much of the track as the player successfully navigated. This includes:

- A starting line, including three lines in the following format:
  - A line consisting of W+2 equals signs (=)
  - A line including:
    - A pipe (|), showing the left edge of the track
    - A series of W spaces, except that the character in column S must be a lowercase letter 'v' to represent the race car's starting position
    - A pipe (|), showing the right edge of the track
  - A line consisting of W+2 dashes (-)
- Lines showing successfully navigated sections of the race track, in the following format:
  - A pipe (|), showing the left edge of the track
  - A series of W characters, including:
    - Spaces indicating clear sections of the track
    - Lowercase letter 'o' to indicate the location of an obstacle
    - Lowercase letter 'v' to indicate the position of the race car
  - Another pipe, showing the right edge of the track
- If the player crashed:
  - A line showing the location of the crash, including:
    - A pipe (|), showing the left edge of the track
    - A series of W characters, including:
      - Spaces indicating clear sections of the track
      - Lowercase letter 'o' to indicate the location of obstacles into which the player did NOT crash
      - Uppercase letter 'X' to indicate the location of the obstacle that was the cause of the crash
    - Another pipe, showing the right edge of the track
  - A line containing the phrase "You Crashed - GAME OVER"
- If the player reached the finish line without crashing:
  - A line consisting of W+2 equals signs (=)
  - A line containing the phrase "Course Complete!"

```
=============
|     v     |
-------------
|     v     |
|    v      |
|   v       |
|o v        |
| vo        |
|v          |
|v          |
|ov         |
|  v        |
|  ov       |
=============
Course Complete!
=============
|      v    |
-------------
|      v    |
|     v     |
|    v      |
|o  v       |
|  X        |
You Crashed - GAME OVER
```