acm International Collegiate Programming Contest 2001

# South Central USA Regional Programming Contest


# Problem #4: Frogger's For Dinner

## Introduction

"Uncle Jacques, " you ask, "What's for dinner?"

"Ask me again in 10 minutes, " Uncle Jacques replies, eyeing the weary-looking frog sitting on the shoulder of Interstate 10, in front of your dilapidated shack.

You notice the potential roadkill as it begins its journey across the vehicle-laden road. You want to know if you should begin boiling a pot of water in anticipation of frog legs for dinner or warm up the leftover possum. You fire up your Swamp 'Puter XL2 and quickly write a program to determine if it is possible for the frog to make it across the road or if it will be hit by a vehicle.

Examining the patch of road in front of your shack, you notice the lanes and shoulders resemble a 10 X 10 grid of squares (shown below). You also notice that the way the frog and the vehicles are moving can be described in "turns". To determine if the frog makes it across the road, you quickly devise a set of rules:

1. At the onset of a run, the frog can start in any square on row 0 (the starting shoulder).
2. At the onset of a run, each vehicle will occupy a square in any column, but only in rows 1-8 (the lanes).
3. Each turn will consist of two steps:
   First, the frog will always remain in the same column and move one row down, towards row 9, his destination (he's not the smartest frog in the world).
   Next, all the vehicles move (at the same time), *n* squares left or right, depending on which row (lane) they are in, where n is their speed (given in the input). To simulate more approaching vehicles, if a vehicle moves off the grid, it instead "wraps around" and appears from the opposite side. Ex: In the grid below, if a vehicle would move to occupy column -1, it would instead occupy column 9 (column -2 would instead occupy column 8, etc.). Also, if a vehicle would move to occupy column 10, it would instead occupy column 0 (column 11 would instead occupy column 1, etc.).

```
     Column
    0123456789
    ----------
R 0|          |<- The frog can start in any square on row 0
o 1|          |(shoulder)
w 2|   /___   |
  3|   \      |cars in rows (lanes) 1-4 move left, or
  4|          |towards column 0
  5|          |
  6|    ___\  |cars in rows (lanes) 5-8 move right, or
  7|       /  |towards column 9
  8|          |
  9|          |<- The destination row (shoulder) of the frog
```

---

4. The frog will succeed in crossing the interstate for a run if it can reach row 9 (without becoming roadkill) after a series of turns starting in ANY column on row 0 (he's not the dumbest frog in the world, either).
5. The frog will become roadkill if at any point it occupies the same square as a vehicle. This includes:
   The frog moving into a square a vehicle occupies, or
   A vehicle "running over" the frog by moving over or into a square the frog occupies.

## Input

Input to this problem will consist of a (non-empty) series of up to 100 data sets. Each data set will describe the starting conditions of the interstate for a run and will be formatted according to the following description. There will be **no blank lines** separating data sets.

1. *Start line* - A single line, "START"
2. The next 8 lines will represent rows 1-8 (the "lanes" of the interstate), starting with row 1. Each line will consist of 10 integers, separated by single spaces. Each integer will represent a column for that row and will be either:
   0, representing no vehicle occupying that square, or
   a non-zero integer *N* in the range 1 <= *N* <= 9, representing a vehicle is occupying that square and the non-zero integer is its speed. NOTE: The given speeds will NOT result in vehicles moving over other vehicles or into a square occupied by another vehicle (no accidents), since all the vehicles move at the same time and all vehicles on a given row are guaranteed to move at the same speed.
3. *End line* - A single line, "END"

## Output

Output for each data set will be exactly one line of output. The line will either be "LEFTOVER POSSUM" or "FROGGER" (both all caps with no whitespace leading or following).

"LEFTOVER POSSUM" will appear if the frog can make it safely (without becoming roadkill) across the interstate after a series of turns starting in ANY column on row 0.

"FROGGER" will be output for a data set if it fails to meet the criteria for a "LEFTOVER POSSUM" line.

## Sample Input

```
START
3 0 0 0 0 3 0 0 0 3
1 0 0 0 1 0 0 0 0 0
4 0 0 0 0 0 0 4 0 0
0 0 2 0 0 0 0 0 0 2
5 0 0 0 0 0 0 0 0 0
0 2 0 0 0 0 0 2 0 2
0 0 0 4 0 0 0 0 0 0
0 2 0 0 0 0 0 0 0 0
END
START
9 9 9 9 9 9 9 9 9 9
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
END
START
1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
END
```

## Sample Output

```
FROGGER
FROGGER
LEFTOVER POSSUM
```