

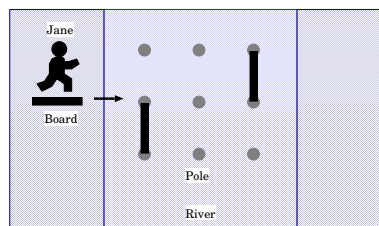
ECE368 Project #5

Due Friday, April 26, 2013, 11:59pm

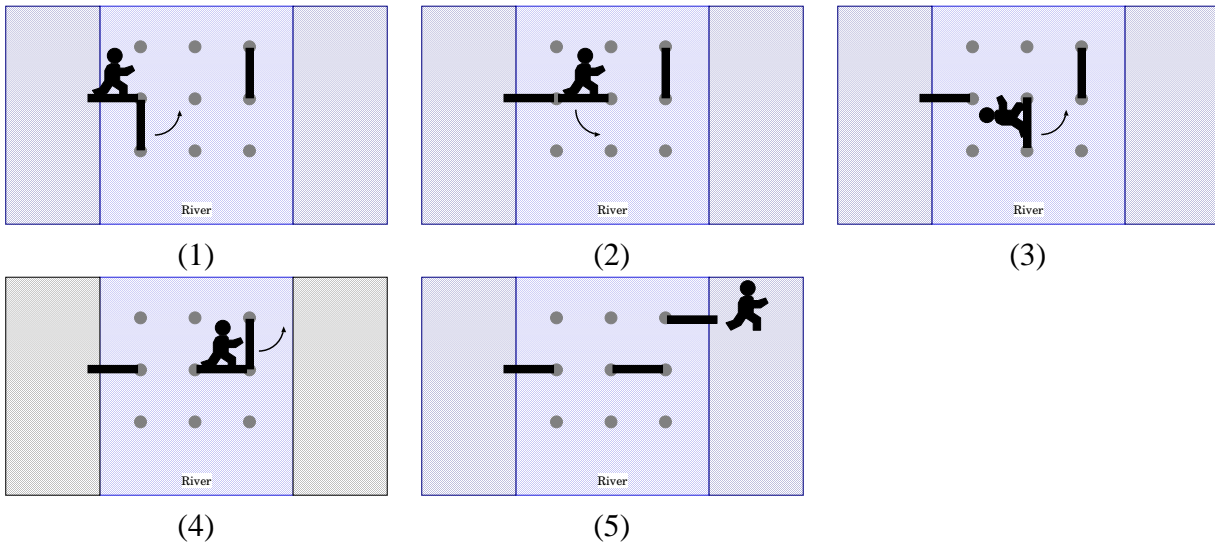
Description

This project is to be completed on your own. This problem is taken from a programming contest organized by the Computer Science Department of the National University of Singapore, at which I spent my sabbatical leave. I was a member of the organizing committee. I did not write this question, but I provided the solution to this problem.

Jane wants to cross a river. The river is dotted with $N \times N$ poles to which boards can latch on. Some of the poles already have boards latched on such that the boards point in the direction of the river flow. For $N = 3$, the situation may look like this.



On her side of the river, Jane has a board that points to the other side. She can latch the board on any of the first column of poles, as shown in (1). She can then use the board as a bridge to reach the pole the board is latched on. When Jane is on a board, she can unlatch one of the two ends of the board and turn the board by 90 degrees such that it latches on another pole, as shown in (1)→(2). Jane can also turn a board by 180 degrees by turning it by 90 degrees twice, as shown in (2)→(3) and (3)→(4). Since turning a board is quite tiresome, she wants to cross the river in such a way that she needs to turn boards by 90 degrees as few times as possible. She does not require any turning to reach any of the first column of poles as she can simply push the board to latch on any of the first column of poles. (However, she may have to turn that board in order to reach a pole in the second column.) In this particular configuration of poles and boards, Jane can cross the river with 4 board turns (5).



Deliverables:

In this project, you are to develop your own include file `river.h` and source file `river.c` (or combine them into one single file), which can be compiled with the following command:

```
gcc -Werror -Wall -Wshadow -O3 river.c -o proj5
```

All declarations and definition of data structures and functions must reside in the include file or source file. The executable `proj5` would be invoked as follows:

```
proj5 input_file
```

The executable loads the configuration of the poles and boards from `input_file`, calculates the number of turns required for Jane to cross the river, and output the number of turns to the `stdout`.

The input file has the following format. The input file should contain N lines, with $N \geq 2$. The first line contains the integer N . Each of the next $N - 1$ lines contains N characters, which are 0 or 1, and these N characters describe the board configuration between two adjacent rows of poles. The character 1 means there is a board and the character 0 means there is no board. Thus the second line of the input file describes the board configuration between the top two rows of poles, and so on, and the last input line describes the board configuration between the bottom two rows of poles. The first of the N characters describes the leftmost board configuration between the leftmost two poles of two adjacent rows of poles, and so on, and the rightmost character describes the rightmost board configuration between the rightmost two poles of the same two rows of poles.

The input file for the starting configuration of the given example contains:

001
100

The output to the stdout should be simply the number 4 followed by a newline.

Grading:

The project requires the submission (through Blackboard) of the C-code (source and include files) and a report. The report should describe the chosen data structure and how you construct the data structure from an given input file. The report should also describe how you compute the number of turns Jane takes to cross the river for a given configuration. Comment on the time complexity of your algorithm (for construction of the chosen data structure and also for the computation of the number of turns required).

Your report should not be longer than 1 page and should be in plain text format or pdf.

Getting started:

Copy over the files from the Blackboard website. Check out the Blackboard website for any updates to these instructions.