A Maze

Write a program that for a given maze (labyrinth) and a given start point will find a path out of the labyrinth.
The labyrinth will be given with 0's and 1's. Where 1 stands for wall, and 0 stands for path, S stands for Start point, F stands for Finish point. The maze won't have loops and will be given as a file to the program.

Example of a maze (input):

```
F0011111111111111111111111111111111111111111111111111111111111111111111111
1101111111110000000001111000000111111111110000111111111111111111111111111
1101111111110111111101111011110000000001111011011111111111111111111111111
1101111111110111111100000011111111111011110110000000000000000000000011111
1101111111110111111111111111111111111011110111111111111111111111111011111
1101111111110111110000001111000000000000000011100000000000000000000011111
1101110000000111110111101111011111111111111111101111111111111111111111111
1101110111111111100111100000011111111111111111000111111111100000000000111
1101000111111111101111111111111110000000001111111011111111110111111110111
1101111111111111101111111111111100111111001111111011111111110000001110111
1101111111110000001111111111111S0001111110011111100111111111111111101110111
1101111110000111101111111111111110000000111111100111111111100000001111111
1101111110111111101111111111111110001111100111100000000000001111111110011
1101111000111111100000011111111111001111000000111111111111101111111000011
1101111111100001101111111111111111000000011110111100001111100001111011111
1101111111101101101111111111111111101111111101111010011111011000000011111
1101111000001100001111111111111111101111111111111010011100011011111111111
1101111011111111111111111111111111101111000111000011011101111000000011111
1101110000000000000000000000001111110000001011001001101110111110111011111
1101110111011111111111111111001111111111111011001101100000110000111000011
1100000111011111111111111111101111111111111000011101111011111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111
```

Here are few of the possible approaches:

1. **Wall follower** (*left-hand rule* or the *right-hand rule*):
   Player keeps one hand in contact with one wall of the maze. In this case the player is guaranteed not to get lost and will reach a different exit if there is one; otherwise, he or she will return to the entrance.
   Good for: mazes with all walls connected and without loops
2. **Tremaux's algorithm**
   is an efficient method to find the way out of a maze that requires drawing lines on the floor to mark a path, and is guaranteed to work for all mazes that have well-defined passages. A path is either unvisited, marked once or marked twice. Every time a direction is chosen it is marked by drawing a line on the floor (from junction to junction). In the beginning a random direction is chosen (if there is more than one). On arriving at a junction that has not been visited before (no other marks), pick a random direction (and mark the path). When arriving at a marked junction and if your current path is marked only once then turn around and walk back (and mark the path a second time). If this is not the case, pick the direction with the fewest marks (and mark it, as always). If there is no exit, this method will take you back to the start where all paths are

marked twice. In this case each path is walked down exactly twice, once in each direction. The resulting walk is called a bidirectional double-tracing.

3. **Shortest path algorithm**
   When a maze has multiple solutions, the solver may want to find the shortest path from start to finish. This algorithm finds the shortest path by implementing a *breadth-first search*. The algorithm uses a *queue* to visit cells in increasing distance order from the start until the finish is reached. Each visited cell needs to keep track of its distance from the start or which adjacent cell nearer to the start caused it to be added to the queue. When the finish location is found, follow the path of cells backwards to the start, which is the shortest path.

For more information about the mazes and algorithms visit:
http://en.wikipedia.org/wiki/Maze
http://en.wikipedia.org/wiki/Maze_solving_algorithm

A possible path:

```
F001111111111111111111111111111111111111111111111111111111111111111111111
110111111110000000001111000000111111111110000111111111111111111111111111
110111111110111111101111011110000000001111011011111111111111111111111111
110111111110111111100000011111111111011110110000000000000000000000011111
110111111110111111111111111111111111101111011111111111111111111111011111
110111111110111110000001111000000000000000011100000000000000000000000011111
110111000000011111011110111101111111111111111110111111111111111111111111
110111011111111110011110000001111111111111111110001111111111000000000111
110100011111111110111111111111111000000000111111101111111111011111110111
110111111111111110111111111111110011111100111111101111111110000001110111
110111111111000001111111111111S000111111001111110011111111111111101110111
110111111000011110111111111111111100000001111111001111111110000000111111
110111111011111110111111111111111100011111001111000000000000111111110011
110111100011111110000011111111111100111100000011111111111110111111000011
110111111110000110111111111111111110000011110111100001111100001111011111
110111111110110110111111111111111111011111110111110100111110110000011111
110111100000110000111111111111111111011111111111110100111000110111111111
110111101111111111111111111111111111011110001110000110111011110000011111
110111000000000000000000000000111111000000101100100110111011111011011111
110111011101111111111111111100111111111111110110011011000001100001110000 11
110000011101111111111111111110111111111111110000111011110111111111111111
111111111111111111111111111111111111111111111111111111111111111111111111
```

For testing purposes don't use the given above maze. Use something simple first, and then the example of the maze given above. For example:

```
11111        111111F11        11F11111        1111111111
11S01        111111011        11000011        1111111111
11101        111111011        11111011        F000001111
111F1 then   111111011  then  11S00011  then  1111100101
             11S100111        11111111        111S110001
             110001111        11111111        1110000111
             111111111        11111111        1111111111
```