Algorithms
Graph Modeling Project (125 pts)

This problem is based on a maze problem (from "MAD MAZES: Intriguing Mind Twisters for Puzzle Buffs, Game Nuts and Other Smart People," by Robert Abbott, Bob Adams, Inc. Publishers, 1990). The description of the problem including a figure is provided separately. You're welcome to try figuring out the solution to the puzzle on your own, but that won't get you any points! Your assignment is to model the maze as a graph and to solve the problem using an appropriate graph algorithm that we've encountered in class. Next, write a program to solve the problem that uses one of the algorithms we covered in Chapter 5 of the textbook. You are encouraged (but not required) to use a library implementation of the graph algorithm.

Important: Note that this is an individual assignment. The departmental collaboration policy (the document you signed at the beginning of the semester) will be strictly enforced.

DELIVERABLES

1. **Problem Modeling[50 pts]**:
   - [10] Explain how you modeled the problem as a graph.
   - [15] Draw enough of the resulting graph to convince us that you have modeled the graph correctly.
   - [10] Identify the graph algorithm needed to solve the problem.
   - [15] Argue that this algorithm will actually solve the problem.

2. **Code Submission [30 pts]**: Your code will first read the input from a file, which is specified as follows: the first line contains a pair of integers representing the number of rows and columns, respectively, in the grid (both are seven in ourexample). Each subsequent line describes the contents of a square in the grid and contains (i) the row number, (ii) the column number, (iii) the *color* of the arrow (R, B), (iv) whether the arrow is Circled or Not (C, N) and (v) the *direction* of the arrow (N, E, W, S, SE, SW, NE, NW). For example, the next input line indicates that the (1,1) square contains an uncircled red arrow pointing to the East. The entire input file is included at the end of this document. Your code will then convert the input to a suitable graph and then run a graph algorithm to compute a solution.

3. **Results [45 pts]**: Display the output of your program on the maze provided. The output should consist of a path from square (1,1) to square (7,7). The path should be described as a sequence of square ids. For example, one possible path may start off as follows

   ```
   (1,1) (1,6) (5,2) (6,2)
   ```

4. **EXTRA CREDIT [10 pts]**: You will receive a bonus of upto 10 points if you can show that you have done something above and beyond what was asked. Possible examples might be (1) using an algorithmic library (e.g., Boost in C++) to implement the graph functionality (2) augmenting your algorithm with a visualization (3) an experimental comparison between algorithms etc.

```
7 7
1 1 R N E
1 2 B N W
1 3 B N NW
1 4 R N NW
1 5 R N S
1 6 B N SW
1 7 B C S
2 1 R N SE
2 2 B N S
2 3 R N SE
2 4 B N S
2 5 R N N
2 6 R N NW
2 7 R N NE
3 1 B N E
3 2 R C N
3 3 R C S
3 4 B N E
3 5 B N N
3 6 B N S
3 7 B N NE
4 1 R N N
4 2 R N NE
4 3 B C E
4 4 R N SW
4 5 R N N
4 6 B N E
4 7 B N SW
5 1 B N W
5 2 R N S
5 3 B C W
5 4 R N SW
5 5 B N S
5 6 B N SE
5 7 R N NE
6 1 R N SW
6 2 B N S
6 3 B N NW
6 4 R C W
6 5 R N SE
6 6 R N S
6 7 B N S
7 1 B N SW
7 2 R N N
7 3 B N E
7 4 R N SW
7 5 B N SE
7 6 B N NW
7 7 X X X
```