You are to write a program that will receive a file name from a user. The filename should be validated(must contain only letters and a period and 3 letters after the period) to ensure it is valid. It will then read in the contents of the file.

The file will contain the size of a 2 dimensional gameboard on its first line and coordinate positions throughout the maze on its second thru ??? lines.

The program should then utilize a class structure to build the gameboard object(Can you say 2 dim dynamic array as a pdm?)

Any coordinates listed in the file will be cells within the gameboard that are inaccessible(walls).

Once the gameboard has been built.

You should write a client program that can contain one additional ADT structure (a Linked List based **StackClass**) which you will use to discover and print **ALL** valid paths through the maze.

**YOU MAY NOT USE RECURSION. My intent is for you to mimic the use of the system stack with our stack class.**

**YOU MAY NOT have any additional lists or queues in the project. You may use as many stacks as needed to solve the problem.**

Valid paths will be printed to file called “solution.out”(named constant w/in your program). List the solution #(starting at 1) on the top of each path that you print. The path’s will be a listing of the directions(not coordinates) that will be traveled to travel from the entrance to the exit of the gameboard.

Players will always enter the game at coordinate[0,0] unless it is blocked.

Players can only exit the game from coordinate [mazeWidth-1, mazeLength-1] and the exit will be only on a downward(southward)movement.

Valid movements within the game are North(up), South(down), East(right) and West(left). These should be represented as an enumerated type and used appropriately in the program.

In any particular path, you may never visit a square more than one time.

A sample data file for a 10x10 maze may be found in the zipped folder. It is called maze.dat.

Deliverables (In order)

CellRec.h - -record for each cell in the maze  
gb.h -- gameboard,h – the maze specification

gb.cpp -- gameboard .cpp --the maze implementation  
stackrec.h -struct for each record in the stack

stack.h -- specification for stack

stack.cpp --implementation for stack

mazeclient.cpp - solves the problem using the classes above Due 4/4