CS 121 **Programming Assignment #4**
Bruce Bolden **20 Points**
March 29, 2023 **Due:** April 7, 2023

Objective Implement a maze traversal program in C++ using *both* dynamic arrays and queues.

Program Description Implement a program to traverse a maze subject to the following constraints:

- Storage for the maze *must be* allocated dynamically.
- Find the path using a queue (positions).

Maze Representation
A maze is represented as a 2D array of characters, where characters are interpreted as follows:

Symbol	Meaning
.	empty space, a place where it is legal to go (move)
#	a wall, a place where it is illegal to move
S	starting position in the maze
G	the goal (final position in the maze)

Maze file format
The mazes to be solved will be stored in a text file, in the following format:
First line: width height (of maze)
Remaining lines: the maze itself, using the characters previously defined.

An example of one possible 5 by 5 maze:

```
5 5
S....
.##.#
.#...
.#.#.
...#G
```

Several mazes *will be* posted on the web site for your program to solve. Your program must solve all of them, no maze will be larger than 40 by 40.

Maze Solution Method
Make the cell at the starting position the current cell. Take the following actions, then repeat:

- If the current cell is adjacent to the exit, stop.
- Mark the current cell as visited.
- Add all *unvisited* neighbors to the north, east, south, and west to a queue.
- Remove the next element from the queue and make it the current cell.

Deliverables

- Submit your programming assignment (design document, programming log, output, and source). It is due at the beginning of class on on Friday, April 7.
- A program design. Describe all data structures and functions necessary to implement your program.
- A complete functional program with output.
- Programming Log:
  - Record the time required to design and implement your program.
  - Record of things you encountered/learned while implementing your program.

Addenda

- Only horizontal and vertical moves are possible. Diagonal moves are not possible.
- Your program must be capable of reading and solving (if possible) a maze as defined in an external file (described above). It should not be necessary to recompile your program to solve a different maze stored in a different file.
- See the web page for additional information. In particular, you should read the page on code style:

  `http://www2.cs.uidaho.edu/~bruceb/General/codeStyle.html`

Grading Scheme

	Maximum Penalty
Source	
Functions	3
Comments	3
Readability	3
Functionality	5
Design	4
Output	4
Programming Log	4