CSC-24400 Programming Project #1
DUE: Tuesday, February 23, 2021, 11:59 PM

1 Objectives

- writing a C++ program on your own
- working with a 2-D array
- utilizing files
- solving a problem using recursion

2 Problem Statement

You will be writing a C++ program that finds a path from the entrance to an exit in a maze (if such an exists in the maze.) Specifically, your program will:

1. prompt the user to enter the name of an input file (that contains no white space.) This input file will contain the details of the maze, including its size, entry and exit points, and actual borders/corridors. The input file will have the following format:

    - the first two items in the input file will be positive integers representing the number of rows r and the number of columns c in the representation of the maze (see below).Note that c and r can be no more than 200.
    - the next two items in the input file will represent the row number and column number of the starting point (i.e. "entrance") in the maze. Remember that row numbers and column numbers both start at 0 (as opposed to 1) in Computer Science.
    - the next two items in the input file will represent the row number and column number of the ending point (i.e. "exit") in the maze. Remember that row numbers and column numbers both start at 0 (as opposed to 1) in Computer Science.
    - the remainder of the input file specifies r rows, each containing c characters. Each character of each row will ether be a `'#'` or a `'+'` character, with a `'#'` representing (part of) a wall/border and a `'+'` representing (part of) a passage/corridor.

Thus, an example input file might be:

```
20 20
10 0
4  19
####################
####################
#####+++++++++++++++#
#####+##############
#####+############+++++
#####++##########++####
#++++##########+####
#+##########+#####+####
#+#####+########+#+####
#+############+#+####
+++++++++++++++#+####
####+##########+####
####+##########+####
#++++##########+####
#+##############+####
#+######+##########+#
#+######+##########+#
#+######+#############
####################
####################
```

2. Your program should next print out the characters in the maze to the screen. That is, it should first print out the original maze essentially as it was specified in the input file.

3. Prints out a few blank rows.

4. Your program should then print out the maze, replacing the `'+'` characters found on the path from the entrance to the exit with `'*'` characters. You should also replace any `'+'` characters that were explored but determined *not* to be part of the solution path with `'B'` characters.

   Quick notes on getting out of a maze from any position that you are in:

   - if you are at the exit you have found a solution.
   - if you are actually at an invalid location (on a border/wall or off of an edge of the maze), this spot cannot be part of a solution. Go back to the spot immediately before you moved here.
   - if you've already been here, then go back to where you came from - there's no point in re-visiting this location.
   - mark the location as (potentially) part of the solution.
   - try the spot to the north (up) from your current location. If there is a way out from there, then there is a way out from here by moving north (up) and then following that way out.
   - if going north didn't work, try a similar idea in each of the other 3 remaining directions (east, south, and west).
   - If none of the 4 directions worked, then mark this spot as not a possible part of the solution and go back to the immediate prior location.

5. finally, if there was not solution, your program should print out the message `NO SOLUTION FOUND`.

3 Example Run

Suppose the file `DeadEnds2.txt` contained the input file specified above. Then one possible execution would be (with user input in *italics*):

`Enter name of input file:` *DeadEnds2.txt*

```
####################
####################
#####+++++++++++++++#
#####+###############
#####+##########++++
#####++##++++++++####
#+++++##########+####
#+##++++++#####+####
#+####+########++#+####
#+############+#+####
+++++++++++++++#+####
####+##########+####
####+##########+####
#++++##########+####
#+##++++++++++++####
#+######+##########+#
#+######+##########+#
#+######++++++++++++#
####################
####################
```

```
####################
####################
#####BBBBBBBBBBBBBB#
#####B###############
#####B##########*****
####BB##+++++++*####
#BBBB##########*####
#B##BBBBBB#####*####
#B####B####BBB#*####
#B############B#*####
*****BBBBBBBBB#*####
####*##########*####
####*##########*####
#+++*##########*####
#+################
#+######+##########+#
#+######+##########+#
#+######++++++++++++#
####################
####################
```

4 What To Hand In

You will be submitting a zip or tgz file containing your source code and a `read.me` file to Canvas. Make sure that you place the project into a single folder (which may contain sub-folders).

The `read.me` file should include information about your project including (but not limited to):

- your name
- the date
- the platform you developed your code on (Windows, Linux, ...)
- any special steps needed to compile your project
- any bugs your program has
- a brief summary of how you approached the problem

You might also want to consider adding things like a "software engineering log" or anything else you utilized while completing the project.

5 Grading Breakdown

Correct Submission	10%
Code Compiles	25%
Correct Execution	50%
Code Formatting/Comments/`read.me`	15%
Early Submission Bonus	*5%*

6 Notes & Warnings

- For most people, this is not the kind of project that you will be able to start the day before it is due and successfully complete. My recommendation is to start *now*.
- If you have any questions about anything involved with this project, ask me. If you don't ask for help, I will not know that you want it. Do not hesitate to stop by my office with questions.
- *Start now*!
- Projects may *not* be worked on in groups or be copied (either in whole or in part) from *anyone or anywhere*. This also means that you may not allow your code to be copied, either in part or in whole, by anyone - such would mean that you (as a provider of code) have also cheated. Failure to abide by this policy will result in disciplinary action(s). See the course syllabus for details.
- *START NOW*!
- Have you started working on this project yet? If not, then *START NOW !!!!!*