Printing Chess Board

Minor Program 1

tl;dr Using functions and a 2D array of characters, students will create a program which reads input from a file representing a chessboard and prints that chessboard to the terminal.

Students will create a program which prompts the user for an input file path. The input file will contain 64 characters representing a chess board. The characters read in are in order from left to right, top to bottom. Below is a reference to what characters represent what pieces in the input.

```
'0' -> Empty Space
'K' -> King
'Q' -> Queen
'B' -> Bishop
'N' -> Knight
'R' -> Rook
'p' -> Pawn
```

Students programs will contain the following functions:

- promptUser() *will print a prompt for the user, get their input file path, and return the path as a string.*
- readFile() *accepts a file path and a 2D array (2D arrays are automatically by reference) and populates the 2D array with characters from the file. It returns nothing.*
- printBoard() *accepts a 2D array meant to represent a chess board and prints it to the terminal.*

Even though the file does not contain characters which distinguish black and white spaces, the function should print non-piece spaces in a chessboard pattern. (where spaces alternate between black/white) Take some time to consider this, there is an easy mathematical approach.

Students can assume that all input files will contain exactly 64 non-whitespace characters. Additionally students may assume that each line will contain exactly 8 non-whitespace characters.

Depending on your method of reading input files, it is possible you may accidentally read newline characters as well. If you run into this issue, simply compare to check for a newline (ex: `if(ch == '\n')`) and then discard/do nothing with that character.

Small Note: I'd really like it if students would use special ascii characters that resemble chess pieces (see starter code) but I know that some systems have trouble displaying these characters. Students are welcome to use simple characters in place of the special ones if it causes them trouble, *it's just much less fun.*

Topics Required

To complete this assignment, students will need to understand C++ functions and 2D arrays.

Purpose

Students completing this assignment will have reviewed and demonstrated earlier concepts from CS1, ensuring they are ready and able to move on to more advanced topics.

Starter code for copy/paste:

```
// Convenient way to use special characters, if your terminal supports it
#define KING "♚"
#define QUEEN "♛"
#define BISHOP "♝"
#define KNIGHT "♞"
#define ROOK "♜"
#define PAWN "♟"
#define DARK_BLOCK "██"
#define LIGHT_BLOCK "  "

#include <iostream>
#include <fstream>
#include <string>

using namespace std;

// Functions must be prototyped
string promptUser();
void readFile(string filepath, string board[8][8]);
void printBoard(string board[8][8]);

int main() { }
```

Formatted with: tohtml.com

Example Input

```
000000N0
00R00000
0000Q000
0p000000
000000p0
0000000B
000K0000
00000000
```

Example Outputs

Output *with* special characters

```
<Input File>: input.txt
  1 2 3 4 5 6 7 8
1  █ █ █♘█
2 █ ♖ █ █ 
3  █ █♕█ █
4 █♙█ █ █ 
5  █ █ █♙█
6 █ █ █ █♗
7  █ ♔ █ █
8 █ █ █ █ 
```

Output *without* special characters

```
<Input File>: input.txt
   1 2 3 4 5 6 7 8
1 ()[]()[]()[]kN[]
2 []()Rk()[]()[]()
3 ()[]()[]Qu[]()[]
4 []pn[]()[]()[]()
5 ()[]()[]()[]pn[]
6 []()[]()[]()[]Bi
7 ()[]()Kg()[]()[]
8 []()[]()[]()[]()
```