CS 142 Assignment 4
Crickets and Grasshoppers
See Canvas for due date!

Crickets and Grasshoppers is a simple two player game played on a strip of *n* spaces. Each space can be empty or have a cricket or grasshopper piece. The first player plays as the cricket pieces and the other plays as grasshoppers and the turns alternate. We'll represent crickets as C and grasshoppers as G. Crickets start on the left side moving right and grasshoppers start on the right side moving left. Each turn, a player **must** move one of their pieces. A piece can either move forward one space if it is empty, or jump over one of the opponent's pieces immediately in front of the jumping piece, landing on an empty space right after the opponent's piece. The number of pieces never changes (pieces are never captured or added). **If no move is possible for the current player, their opponent wins the game.**

Begin by asking how many pieces each player has with a prompt using `System.out.print` like the following, **with 10 as the maximally allowed number of pieces** (user input shown **bold and underlined**):

```
Please enter the number of pieces for each player (1-10): two
```

To do this, use the `promptNumberReadLine` method that you will write, described on the back page. If the user does not type a number in the correct range, that method will prompt them in this way:

```
That was not a valid number! Please try again.
Please enter the number of pieces for each player (1-10): 2
```

Next, ask how many spaces should be in the middle in the following way with 9 as the maximum:

```
Please enter the number of spaces in the middle (1-9): 1
```

The strip of *n* squares consists of the specified pieces for each player on their respective ends with the given empty spaces in the middle. Before each move, display the current state of the game by printing out the result of `boardToString`, then ask the player which position to move (1 through *n*) using the `promptNumberReadLine` method. Please also note the error messages and re-prompting below:

```
CC.GG
Crickets, please enter a position to move (1-5): 1
That space does not contain a piece you can move! Please try again.
Crickets, please enter a position to move (1-5): 4
That space does not contain a piece you can move! Please try again.
Crickets, please enter a position to move (1-5): 7
That was not a valid number! Please try again.
Crickets, please enter a position to move (1-5): 2
C.CGG
Grasshoppers, please enter your move (1-5): 4
CGC.G
Crickets, please enter your move (1-5):
```

During the first move, only positions 1 and 2 contain the cricket pieces (and only the piece in position 2 can be moved), but as moves are made, other positions could contain crickets, so we always include all position numbers in the prompt. Notice that the same "does not contain" error message is given if a position that can't move is entered, whether that position contains a trapped player piece, an opponent piece, or a space. However, the "valid number" error message is given if the number was not in the correct range. The `promptNumberReadLine` method will only provide the second error message.

Please create a class called **CricketsAndGrasshoppers** that implements the methods on page 2.

Print the `prompt` using `System.out.print(…)`. If the next piece of information in the Scanner represents an integer which is at least 1 and at most `max`, return the number, but make sure the Scanner also **reads in the rest of the line** before returning. Otherwise (if the next piece of information doesn't represent the above requirements), **read in the rest of the line**, print the line
`That was not a valid number! Please try again.`
and repeat the process described in this box. **Follow these directions precisely for full points.**
`public static int promptNumberReadLine(Scanner s, String prompt, int max)`

Create and return an array representing a new game with the number of pieces for each player indicated. The pieces should be on the ends of the board with the specified empty spaces.
`public static int[] createBoard(int piecesPerPlayer, int spacesInMiddle)`

Create and **return** a String that represents the game board, all on one line. **Don't print it!** Crickets are specified with C, grasshoppers with G, and empty spaces with . (period) **Hint:** Use string concatenation +
`public static String boardToString(int[] board)`

Return true if the given player has any move they can make. Cricket is player 1 and grasshopper is 2.
`public static boolean canMove(int[] board, int player)`

The player moves their piece in the given position **(numbered 1 through *n*)**. If the specified move is allowed, modify the board and return true. Otherwise, don't modify board and return false. **Don't print!**
`public static boolean move(int[] board, int player, int position)`

Write a main method which creates the Crickets and Grasshoppers game with the wording provided.
`public static void main(String[] args)`

You may write additional helper methods if desired. **Only the promptNumberReadLine and main methods should print.** Here are three example runs of `main` (user input shown **bold and underlined**):

```
Please enter the number of pieces for each
player (1-10): 2
Please enter the number of spaces in the
middle (1-9): 1
CC.GG
Crickets, please enter your move (1-5): 2
C.CGG
Grasshoppers, please enter your move (1-5): 4
CGC.G
Crickets, please enter your move (1-5): 3
CG.CG
Grasshoppers, please enter your move (1-5): 5
CGGC.
Crickets, please enter your move (1-5): 4
Crickets win!
```

```
Please enter the number of pieces for each
player (1-10): 2
Please enter the number of spaces in the
middle (1-9): 2
CC..GG
Crickets, please enter your move (1-6): 2
C.C.GG
Grasshoppers, please enter your move (1-6): 5
C.CG.G
Crickets, please enter your move (1-6): 1
.CCG.G
Grasshoppers, please enter your move (1-6): 5
That space does not contain a piece you can
move! Please try again.
Grasshoppers, please enter your move (1-6): 6
Grasshoppers win!
```

```
Please enter the number of pieces for each
player (1-10): 2
Please enter the number of spaces in the
middle (1-9): 2
CC..GG
Crickets, please enter your move (1-6): 2
C.C.GG
Grasshoppers, please enter your move (1-6): 5
C.CG.G
Crickets, please enter your move (1-6): 3
C..GCG
Grasshoppers, please enter your move (1-6): 4
C.G.CG
Crickets, please enter your move (1-6): 1
.CG.CG
Grasshoppers, please enter your move (1-6): 0
That was not a valid number! Please try again.
Grasshoppers, please enter your move (1-6): 3
GC..CG
Crickets, please enter your move (1-6): 2
G.C.CG
Grasshoppers, please enter your move (1-6): 6
G.CGC.
Crickets, please enter your move (1-6): 5
G.CG.C
Grasshoppers, please enter your move (1-6): 4
GGC..C
Crickets, please enter your move (1-6): 3
Crickets win!
```