CSCI 14 Programming assignment #8 (more functions) -- 40 points. Due 4/10/18

Recall the diamond assignment from #6 and #7. You will now add multiple shapes, the original diamond, a triangle, a parallelogram, a pyramid, and an hourglass. Add a function `GetShape()` that puts up a menu of shapes and forces the user to pick a shape (by number) or ask to quit (another number). Force an entry from your menu's range of options, and return that value to the caller. Your `main()` function will run in a small loop – call `GetShape()` until it returns "quit" and call `DrawShape()` each time the user hasn't told you to quit.

You may not use any global variables: declare all variables within the appropriate functions, and pass them around as needed.

You may not add any parameters to any of these functions besides what I give here. The major logic change will be in the functions to calculate the number of spaces and stars for the specific shape the user asked for this pass. Most of the rest of the program will basically not change. You will add the shape parameter only to those functions that need it. The `GetNRows()` function needs it, since the number of rows only has to be odd for the shapes that have a "top" and "bottom" half, but not for shapes that are not symmetric about the center row.

```
// get the shape or 'quit' command returning the user's option
int GetShape( void );

// get number of rows for that shape, assuring legal, and return
it
int GetNRows( int shape );

// draw the shape, using helpers as needed (described below)
void DrawShape( int shape, int nRows );

// calculate number of spaces given current row and row number
// was nSpaces(), this is a better function name (verb-like)
int CalcNSpaces( int shape, int nRows, int row );

// calculate number of stars given current row and row number
// was nStars(), this is a better function name (verb-like)
int CalcNStars(int shape, int nRows, int row );

// print n copies of character c on the current row
void PrintChars( int n, char c );
```

The diamond remains the same. A 4-row triangle looks like this:

```
****
 ***
  **
   *
```

A 5-row parallelogram looks like this:

```
   *
  **
 ***
**
*
```

A 6-row pyramid looks like this:

```
     *
    ***
   *****
  *******
 *********
***********
```

A 5-row hourglass looks like this:

```
*****
 ***
  *
 ***
*****
```

Use the following numbers for the shapes menu:
1 is diamond
2 is triangle
3 is parallelogram
4 is pyramid
5 is hourglass
6 is the quit command.