COSC 1436, Dr. Pershwitz

Assignment 5b: 9 points

Write a Program: Print a table of stars and zeroes

Your goal is to print a square table of alternating stars and zeroes.

Ask the user to enter a size of the table. This number should be odd. Since the table is square the number of rows and columns will be the same and should be equal to the entered size. The valid range for the size is from 3 to 15, inclusive.

Use a loop to make sure the entered size is odd and within the range. Output a descriptive error message if an invalid size is entered. The error message should indicate if the size is even, or less than the minimum, or larger than the maximum. If more than one error condition apples, e.g. the entered number is even and larger than the maximum, then both error conditions need to be printed. Do not create separate output strings for all combinations of error conditions – your program should build the output.

Once a valid size is entered print the table to the console. Each row of the table should have alternating '*' and '0' symbols separated by a space. Each column of the table should also have alternating '*' and '0' symbols. All corners should have the '*' symbol. Please see the output examples. **Hint:** a trailing space in each row is okay.

Your program should have the following:

- You should have 4 comment lines at the top: description of the program, author, section, and date. (1 point)
- Create your variables, use the appropriate type, name them appropriately, and remember to not leave them uninitialized. (1 point)
- Make sure the scope of your variables is as small as possible and does not extend beyond the blocks of code where they are needed (1 point)
- You should define appropriate named constants and name them appropriately. (1 point)
- Ask the user to enter a size of the table and read it from the console. If an invalid size is entered output an error message and ask to enter a valid size. Your error message should indicate what constraint(s) on the size is(are) violated. (2 points)
- Once you get a valid size use nested loops to print the table (see example output). (3 points)

Your output should look similar to the following:

First run

```
enter an odd size from 3 to 15: 10
error: that's even!
enter a valid size: 1
error: that's less than the min!
enter a valid size: 17
error: that's greater than the max!
enter a valid size: 2
error: that's even! that's less than the min!
enter a valid size: 16
error: that's even! that's greater than the max!
enter a valid size: 3

* 0 *
0 * 0
* 0 *
```

Second run

```
enter an odd size from 3 to 15: 13

* 0 * 0 * 0 * 0 * 0 * 0 *
0 * 0 * 0 * 0 * 0 * 0 * 0
* 0 * 0 * 0 * 0 * 0 * 0 *
0 * 0 * 0 * 0 * 0 * 0 * 0
* 0 * 0 * 0 * 0 * 0 * 0 *
0 * 0 * 0 * 0 * 0 * 0 * 0
* 0 * 0 * 0 * 0 * 0 * 0 *
0 * 0 * 0 * 0 * 0 * 0 * 0
* 0 * 0 * 0 * 0 * 0 * 0 *
0 * 0 * 0 * 0 * 0 * 0 * 0
* 0 * 0 * 0 * 0 * 0 * 0 *
0 * 0 * 0 * 0 * 0 * 0 * 0
* 0 * 0 * 0 * 0 * 0 * 0 *
```

Notes:

- When you run your program, you should test both sides of the valid range for even and odd sizes. You should also test if the size is within the range and even.
- Please include two runs with different table sizes. You don’t have to test the validity of input the second time. Insufficient console output is a 1-point deduction.
- Pay attention to where you create and how you initialize your variables. Unsafe code is a 1-point deduction.
- Comment your code. Uncommented code is also a 1-point deduction.

Extra credit (4 points max):

- You can earn 2 points of extra credit if you surround your table with a border using the ‘-’ and the ‘|’ symbols. You still need to have two runs and test invalid input as specified in the notes.

Example extra credit output:

```
enter an odd size from 3 to 15: 13
-----------------------------
| * 0 * 0 * 0 * 0 * 0 * 0 * |
| 0 * 0 * 0 * 0 * 0 * 0 * 0 |
| * 0 * 0 * 0 * 0 * 0 * 0 * |
| 0 * 0 * 0 * 0 * 0 * 0 * 0 |
| * 0 * 0 * 0 * 0 * 0 * 0 * |
| 0 * 0 * 0 * 0 * 0 * 0 * 0 |
| * 0 * 0 * 0 * 0 * 0 * 0 * |
| 0 * 0 * 0 * 0 * 0 * 0 * 0 |
| * 0 * 0 * 0 * 0 * 0 * 0 * |
| 0 * 0 * 0 * 0 * 0 * 0 * 0 |
| * 0 * 0 * 0 * 0 * 0 * 0 * |
| 0 * 0 * 0 * 0 * 0 * 0 * 0 |
| * 0 * 0 * 0 * 0 * 0 * 0 * |
-----------------------------
```

- You can earn another point of extra credit if you appropriately use the conditional (ternary) operator in your program. This will go to 2 points if you don’t have a single `if` statement in your program.