CSCI 1470 **Fall 2024** **Mr. Gustavo Dietrich**

Lab 19: Two-Dimensional Arrays **Due:** 11/29/24

In this lab you will practice working with two-dimensional arrays.

Example Program

This program uses nested loops to process a squared two-dimensional array (has the same number of rows as columns) to represent a grid of two-digit numbers. First the array is filled with numbers generated so that their first and second digits correspond to the row and column respectively where the number is in the grid. The array is then passed to a function that displays as many rows and columns as specified by the user. Analyze and understand this program before you start writing your program. It will help you a lot.

Your Program

Write a C++ program that creates a checkerboard like the one shown in Figure 1. It then prints the board as is (Figure 1) and rotated like the one shown in Figure 2.
Your program must fulfill the following requirements:
1) You must use a **7x7** (7 rows by 7 columns) **two**-dimensional **char** array to implement the board.
2) You must use a **global** constant variable to specify the dimensions (rows and columns) of the array.
3) You must use nested loops to solve the problem.
4) Function **main()** must:
    a) Create the **array.**
    b) Allow the user to create as many checkerboards as desired.
    c) Prompt the user to enter the size of the board (quantity of rows and columns) that he/she wants to create. The size must be within the range 1-7.
    d) Call the three functions specified below to create and print the board.
5) Function **createBoard()**: receives the array and the size of the checkerboard specified by the user from main() and returns it filled with the checkerboard.
6) Function **printBoard1()**: receives the array and the size of the checkerboard specified by the user from main() and prints it as is (see Figure 1).
7) Function **printBoard2()**: receives the array and the size of the checkerboard specified by the user from main() and prints it rotated (see Figure 2).

Example:

```
A   B   C
  D   E   F
G   H   I
  J   K   L
M   N   O
  P   Q   R
```

Figure 1

```
A   G   M
  D   J   P
B   H   N
  E   K   Q
C   I   O
  F   L   R
```

Figure 2

Note: Here the checkerboards are shown side by side to facilitate the comparison. The actual output will show one on top of the other.
The rows and columns are highlighted just to show they have to be rotated.

Open **lab19.cpp** in your IDE and implement the algorithm provided in the source code as comments according to the above specifications.

Tips:

1) You can assign to a **char** variable the ASCII code of the character that you want to store in it and do arithmetic with it. For example:

```
char x = 97;        // assigns ASCII code corresponding to 'a'
cout << x;          // displays 'a'
++x;                // increments the value in x (that is, it becomes 98, ASCII code corresponding to 'b')
cout << x;          // displays 'b'
```

2) Once you figured out how to create the checkerboard, printing it as in Figure 1 is pretty straight forward. To print it like in Figure 2 you need to make minor changes.

Important:

- You must choose the **most** appropriate data types for your variables, parameters, and functions.
- When you pass an array to a function make sure you pass it in the most appropriate way (const or not const).
- See below sample runs showing the behavior of my program for valid and invalid entries. Pay close attention to them to see what your program must do.

Please run your program to ensure that you tested it. Make sure it behaves like my sample solution.

Don't forget to include at the top of the program the comments shown below with your information (name, class and section number, etc.)

```
////////////////////////////////////////////////////////////////////
//
// Name: <Put your name here>
// Date: <Today's date>
// Class: <Your class number and section number, like: CSCI 1470.02>
// Semester: <This semester, like: Fall 2012>
// CSCI/CMPE 1470 Instructor: <Your lecture instructor's name>
//
// Program Description: Enter here your description of what the program does
//
////////////////////////////////////////////////////////////////////
```

When done, submit your solution through Blackboard using the "Assignments" tool. Do Not email it.

Paste the link to your final solution along with your source code in the textbox opened when you click on Create Submission before you click on Submit.

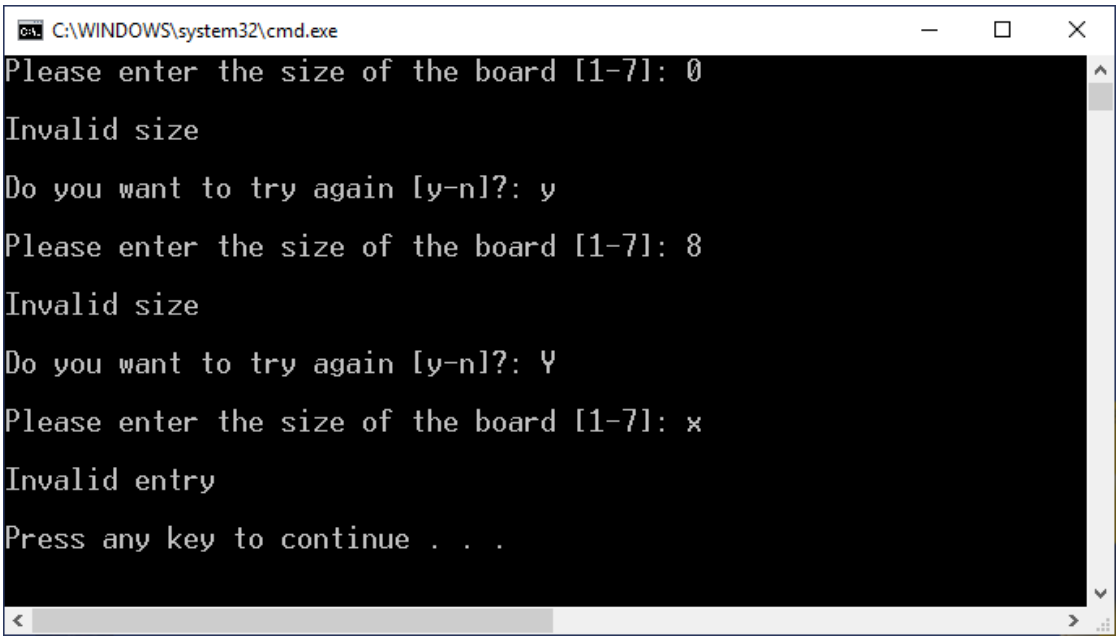
The following is the basic criteria to be used to grade your submission:

You start with 100 points and then lose points as you don't do something that is required.

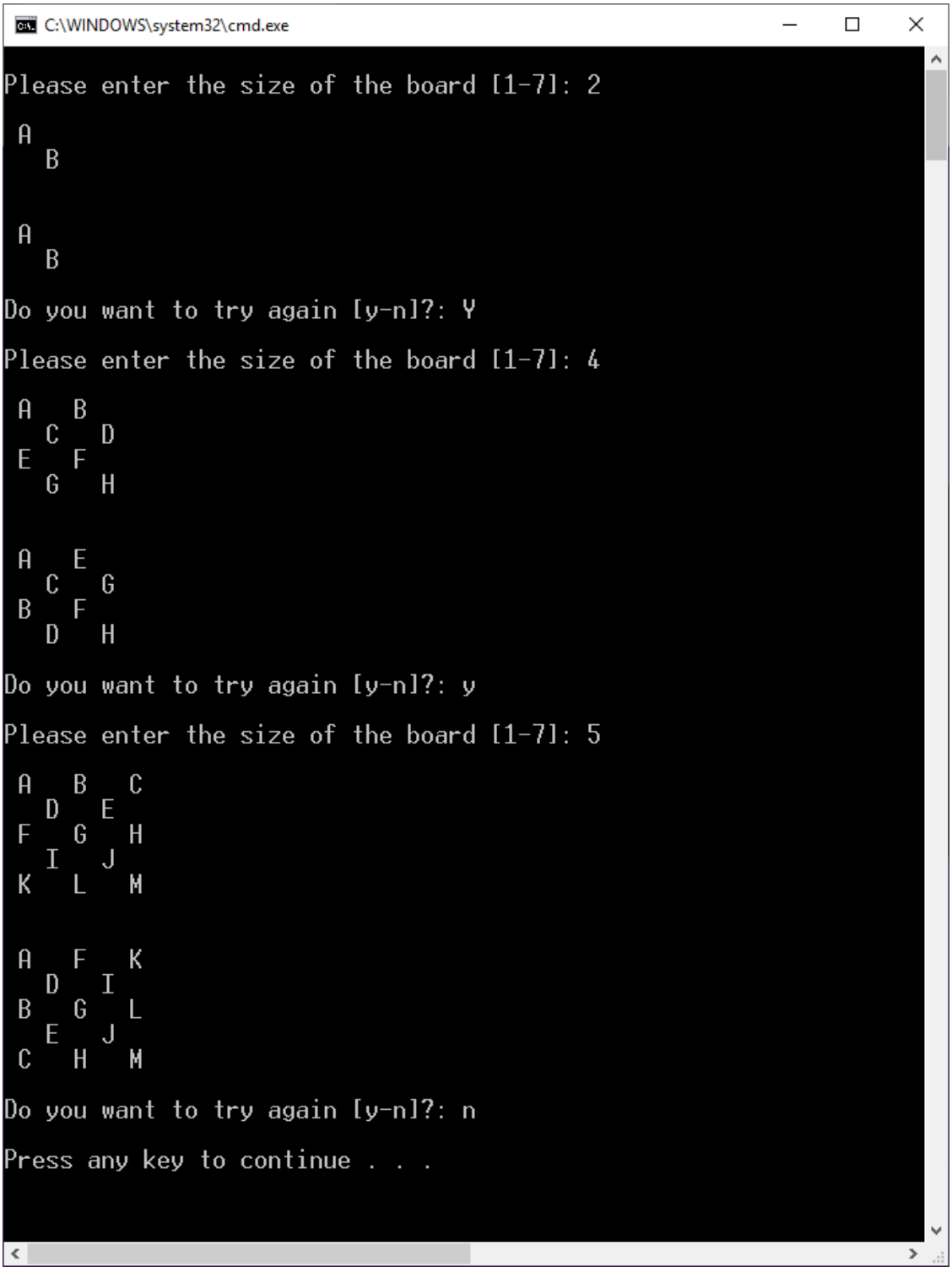
-40: Doesn't use a two-dimensional array
-40: Doesn't compile
-30: Doesn't run correctly
-5: Arrays are not passed to the functions properly (const/not const)
-5: Incorrect file name
-10: **main()** doesn't properly check input stream
-10: **main()** doesn't properly check the range for bsize
-5: **main()** doesn't properly display error messages
-15: **main()** doesn't properly implement the loop
-10: **createBoard()** doesn't properly create the checkerboard
-10: **printBoard1() d**oesn't properly print the checkerboard
-10: **printBoard2() d**oesn't properly print the rotated checkerboard
-15: Missing function (each)
-10: incorrect parameter list (each function)
-10: incorrect function call (each function)
-20: program does not implement the provided algorithm
-20: Incorrect/missing cpp file or incorrect/missing link to your solution
-100: The code submitted is not your creation (you got it from a web site or another person)
-10: Late

Note: more points may be lost for reasons not specified here.

Sample runs of my program with invalid entries



Sample runs of my program with valid entries


C:\WINDOWS\system32\cmd.exe
Please enter the size of the board [1-7]: 2
A
B
A
B
Do you want to try again [y-n]?: Y
Please enter the size of the board [1-7]: 4
A B
C D
E F
G H
A E
C G
B F
D H
Do you want to try again [y-n]?: y
Please enter the size of the board [1-7]: 5
A B C
D E
F G H
I J
K L M
A F K
D I
B G L
E J
C H M
Do you want to try again [y-n]?: n
Press any key to continue . . .