EECE.2160: ECE Application Programming
Fall 2017

Programming Assignment #7: Simple Raster-Scan Graphics
Due **Friday, 4/13/17**, 11:59:59 PM

1. Introduction

This assignment will give you practice working with two-dimensional arrays by implementing a simple raster-scan graphics program. In raster-scan graphics, drawings are constructed one pixel at a time, sweeping from left-to-right to print a line and printing the lines from top to bottom. You will use a two-dimensional array to store a grid and a series of boxes added to that grid using string-based commands.

This assignment is adapted from an earlier assignment written by Prof. George Cheney.

2. Deliverables

This assignment uses multiple files (note: starter versions of each file are available on the course website):

- ***prog7_raster.c:*** Source file containing your main function.

- ***prog7_functions.h:*** Header file containing function prototypes..

- ***prog7_functions.c:*** Source file containing other user-defined functions

Submit all three files by uploading these files to your Dropbox folder. Ensure your file names match the names specified above. Failure to meet this specification will reduce your grade, as described in the program grading guidelines.

3. Specifications

Basic setup: All program operations manipulate the state of a 21 x 51 "pixel" grid, the initial state of which is shown in Figure 1:
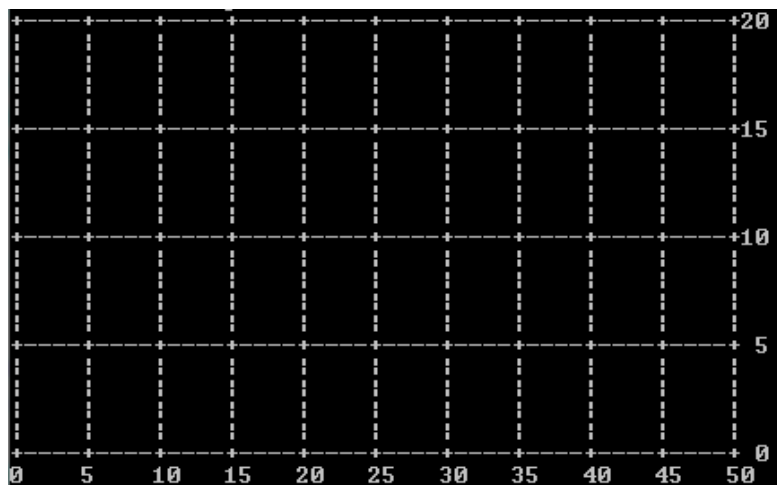

Figure 1: Empty 21 x 51 grid

Note that:

- The x-axis runs from 0 to 50; the y-axis runs from 0 to 20.
- The state of the grid is stored in a 2-dimensional character array declared in main(), with each character representing one of the following:
    - An intersection of two grid lines (`'+'`)
    - A horizontal grid line (`'-'`)
    - A vertical grid line (`'|'`)
    - A space between grid lines (`' '`)
- The axis labels shown on the right and bottom of the grid are not stored in this array; they are simply printed when the current grid state is printed.
- To change the display, users can add "boxes" to the grid by specifying the (x,y) coordinates of the origin (lower left corner), the width, and the height.
    - Each point in each box is displayed as a star (`'*'`), which overwrites all appropriate positions in the grid. Figure 2 shows a grid with 4 boxes.
    - Note that boxes can lie partially outside the grid—you must determine what part of the box is inside the grid and store those coordinates. The box in the top right corner of Figure 2 shows one example.
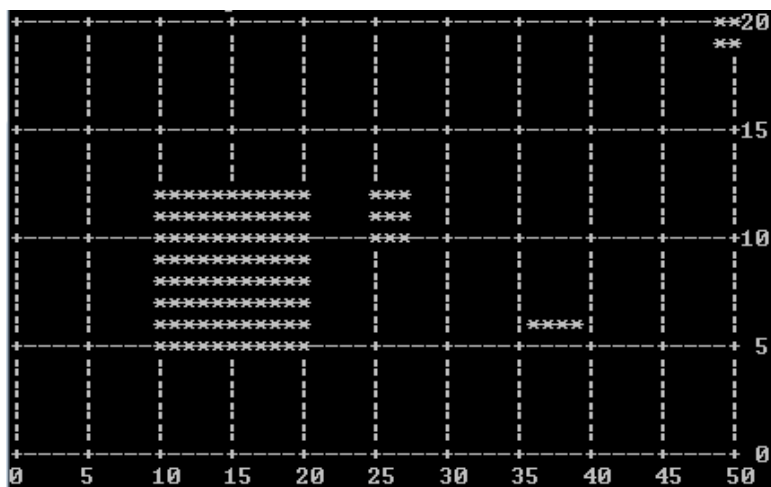
Figure 2: Updated grid with the following boxes—origin (10,5) with width 11 and height 8, origin (25,10) with width and height 3, origin (36,6) with width 4 and height 1, and origin (49,19) with width and height 4. Note that the fourth box goes outside the grid, but only the lower left part of it is shown.

Input: Your program should handle each of the following commands:

- `add`: Add a box to the grid--prompt the user to enter the x and y coordinates of the origin, as well as the width and height.
- `print`: Print the current state of the grid and all added boxes.
- `reset`: Remove all boxes from the grid and reset it to its initial state.
- `exit`: End the program.

If the user enters any other command, print an error message.

Output: When the user enters the print command, print the current state of the grid and all added boxes, as shown in the above examples and in the test cases below.

Error checking: Your program needs only to check for invalid commands; you do not need to perform any additional error checking.

Again, note that a box that is partially outside the grid is <u>not</u> an error condition—you simply need to account for the part of the box that is inside the grid. If a box is completely outside the grid, you can ignore it without printing any errors.

Functions: In addition to the `main()` function, I suggest using the following additional functions, which are called by `main()`:

- `void resetGrid(char grid[][NCols])`: Reset the grid to its initial state by removing all boxes and resetting all characters to gridlines or spaces.

- `void addBox(char grid[][NCols], int x, int y, int width, int height)`: Add a box to the grid with origin (x,y) and the specified width and height. Note that the actual console input should be done outside this function, and that this function should account for cases where part of the box is outside the grid.

- `void printGrid(char grid[][NCols])`: Print the current state of the grid, including all added boxes.

4. Test Cases

Your output should match these test cases exactly for the given input values. I will use these test cases in grading of your lab, but will also generate additional cases that will not be publicly available. Note that these test cases may not cover all possible program outcomes. You should create your own tests to help debug your code and ensure proper operation for all possible inputs.

I've copied and pasted the output from each test case below, rather than showing a screenshot of the output window. User input is underlined in each test case, but it won't be when you run the program.

```
Enter command: add
Enter X and Y coordinates for origin: 2 3
Enter width and height: 5 10

Enter command: print
+----+----+----+----+----+----+----+----+----+----+20
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+----+15
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
| *****    |    |    |    |    |    |    |    |    |
| *****    |    |    |    |    |    |    |    |    |
+-*****---+----+----+----+----+----+----+----+----+10
| *****    |    |    |    |    |    |    |    |    |
| *****    |    |    |    |    |    |    |    |    |
| *****    |    |    |    |    |    |    |    |    |
| *****    |    |    |    |    |    |    |    |    |
+-*****---+----+----+----+----+----+----+----+----+ 5
| *****    |    |    |    |    |    |    |    |    |
| *****    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+----+ 0
0    5   10   15   20   25   30   35   40   45   50

Enter command: add
Enter X and Y coordinates for origin: -3 -3
Enter width and height: 10 10

Enter command: add
Enter X and Y coordinates for origin: 40 15
Enter width and height: 3 6
```

4. Test Cases (continued)

```
Enter command: print
+----+----+----+----+----+----+----+----+---***--+----+20
|    |    |    |    |    |    |    |    ***  |    |
|    |    |    |    |    |    |    |    ***  |    |
|    |    |    |    |    |    |    |    ***  |    |
|    |    |    |    |    |    |    |    ***  |    |
+----+----+----+----+----+----+----+----+---***--+----+15
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|  *****   |    |    |    |    |    |    |    |    |
|  *****   |    |    |    |    |    |    |    |    |
+-*****---+----+----+----+----+----+----+----+----+10
|  *****   |    |    |    |    |    |    |    |    |
|  *****   |    |    |    |    |    |    |    |    |
|  *****   |    |    |    |    |    |    |    |    |
*******    |    |    |    |    |    |    |    |    |
*******---+----+----+----+----+----+----+----+----+ 5
*******    |    |    |    |    |    |    |    |    |
*******    |    |    |    |    |    |    |    |    |
*******    |    |    |    |    |    |    |    |    |
*******    |    |    |    |    |    |    |    |    |
*******---+----+----+----+----+----+----+----+----+ 0
0    5   10   15   20   25   30   35   40   45   50

Enter command: clear
Invalid command clear

Enter command: reset
```

5

4. Test Cases (continued)
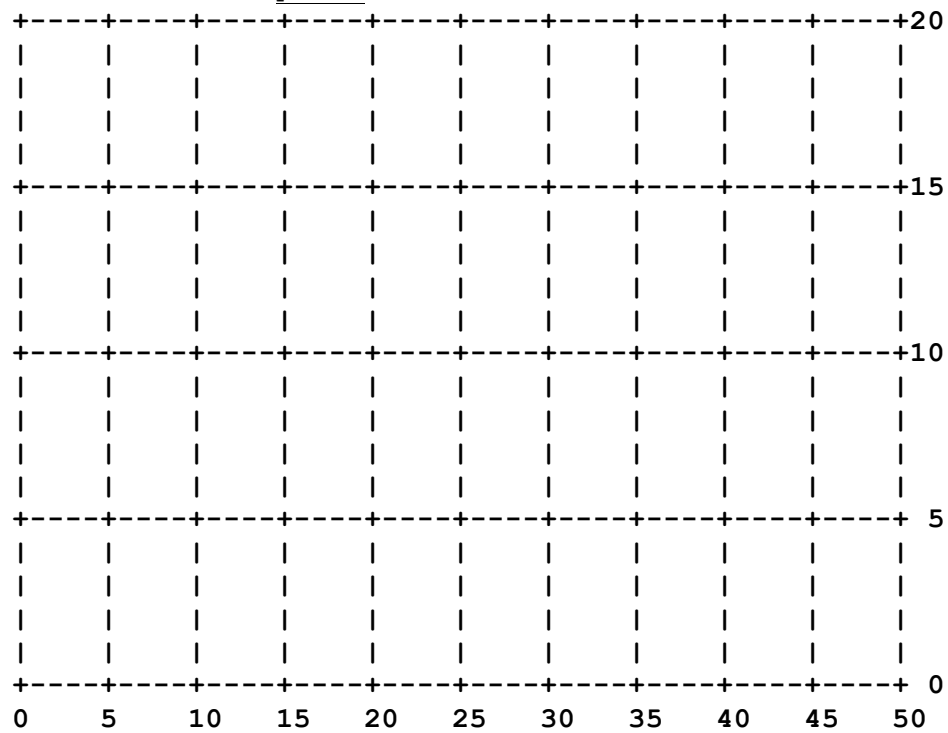
```
Enter command: print
+----+----+----+----+----+----+----+----+----+----+20
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+----+15
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+----+10
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+----+ 5
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+----+ 0
0    5    10   15   20   25   30   35   40   45   50

Enter command: quit
Invalid command quit

Enter command: exit
```