CS 222: Systems Programming

Homework #4: Simple Raster-Scan Graphics

Due **Monday of Week 14**

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.

2. Deliverables

This assignment uses multiple files and the starter version is available on Canvas:

- ***hw4_raster.c:*** Source file containing your main function.
- ***hw4_functions.h:*** Header file containing function prototypes.
- ***hw4_functions.c:*** Source file containing other user-defined functions
- ***Makefile:*** Write your own file.

Submit all four files by uploading these files to Canvas as a zipped archive – ***hw4.zip***. Ensure your file names match the names specified above.

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:

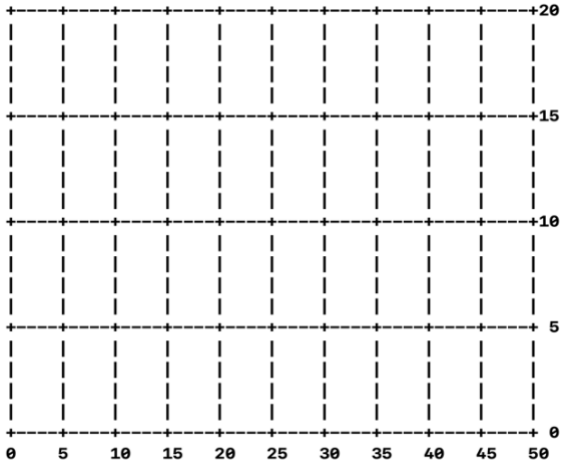


Figure 1: An empty 21 x 51 grid

Note:

- 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.

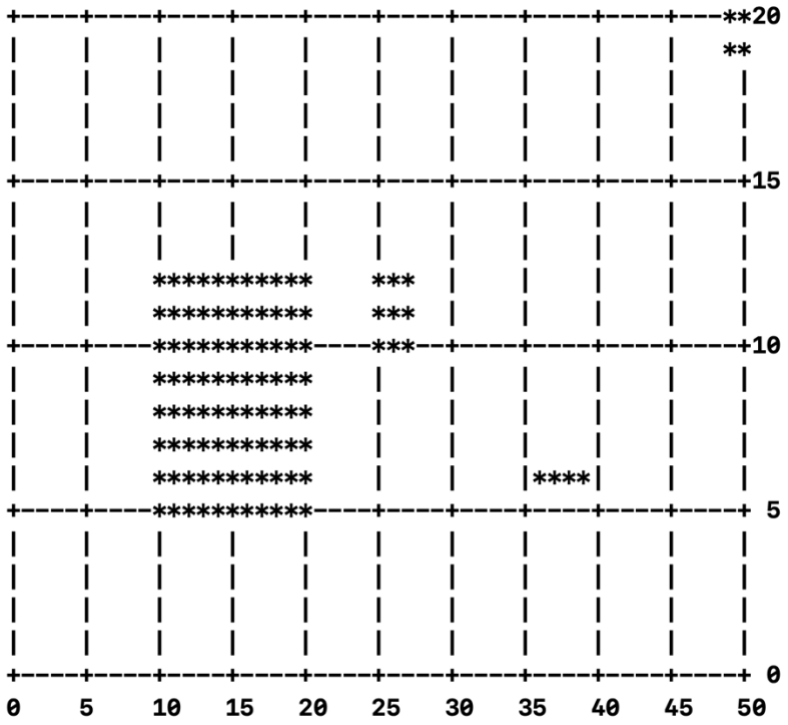


Figure 2: Updated grid with the following boxes—origin (10,5) with width 11 and height 8, origin (25,10) with width and height 3, origin (36,6) with width 4 and height 1, and origin (49,19) with width and height 4. **Note** that the fourth box goes outside the grid, but only the lower left part of it 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 not 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, you must complete 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

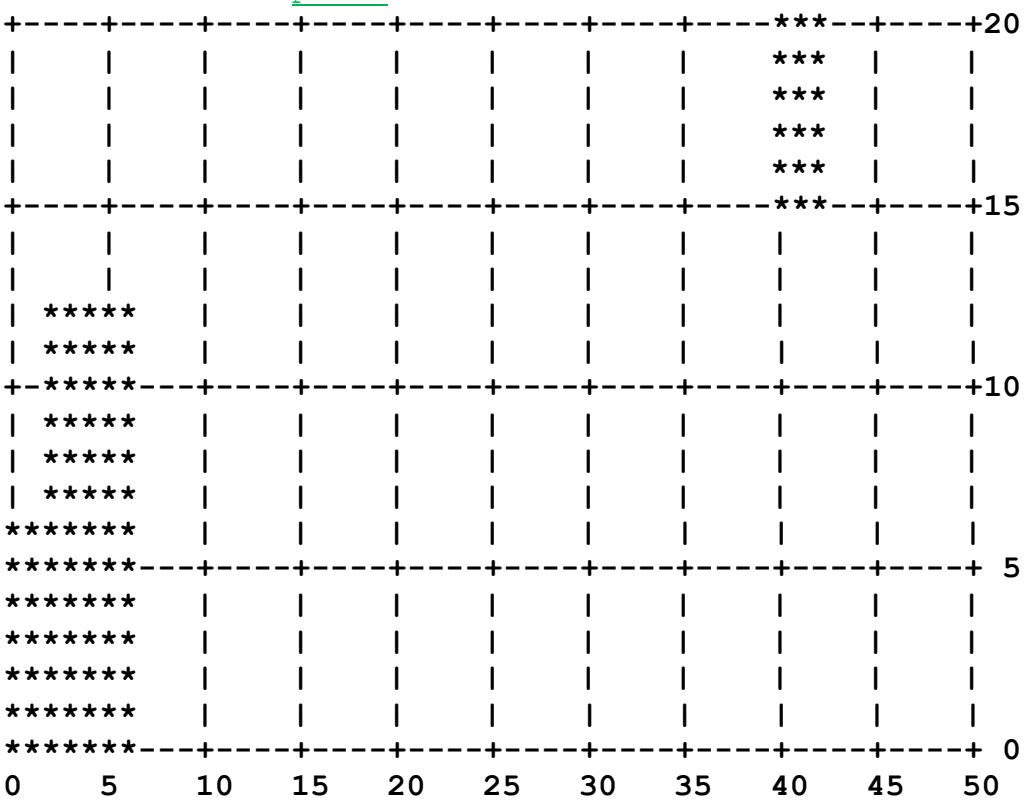
Your output should match these test cases exactly for the given input values. I will use these test cases in grading of your lab, but will also generate additional cases that will not be publicly available. Note that these test cases may not cover all possible program outcomes. You should create your 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 and color-coded in green 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
```

4. Test Cases (continued)

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

Enter command: print
```




```
Enter command: clear
Invalid command clear

Enter command: reset
```

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
```

5. Deductions

Deduction Code	Description	Points Deducted
1	Program does not compile, but some code was given.	-60
2	Comments	-10
2a	No header comments	-4
2b	Body of program contains no single line of comment	-4
2c	Other comments such as comments for key variables. Header comments is present but missing some key components	-2
3	File names are incorrect (including source files and Makefile)	-10
4	Inconsistent program style • Indentation, braces, descriptive variable names, etc. • -2 for each occurrence and maxed at -10	-10
5	Input error	-10
5a	Program can intepret more commands than the given five. (-5)	-5
5b	Program cannot interpret some commands. (-5)	-5
6	Output error	-50
6a	Output error due to incorrect resetGrid() function i. Grid is mis-shaped when resetting. (-5) ii. Characters in the grid is incorrect when resetting. (-5)	-10
6b	Output error due to incorrect addBox() function i. Function does not handle partial boxes on the grid border correctly. (-5) ii. Boxes are offset when added. (-5) iii. 2D array not updated by the function. (-10)	-20
6c	Output error due to incorrect main() function (-5 for each until maxed) i. Incorrect calling on the functions under each command	-10
6d	Printed information does not match with the given sample output exactly (-2 for each occurrence until maxed) i. There should be no extra information printed ii. User prompts and error messages should match	-10
7	Lack of a Makefile or Makefile cannot compile the code	-10
7a	Makefile does not have a recipe for cleaning the object files and executable.	-5
7b	Makefile does not separate the compilation and linking steps.	-5