ECE 109 Program 3: school.asm

Spring 2020 Updated 4/3/20

Due April 22, 2020 @ 11:45pm

This programming assignment must be completed individually. Do not share your code with or receive code from any other student. The only people who may see your code are the instructor and the ECE 109 TAs.

Evidence of copying or other unauthorized collaboration will be investigated as a potential academic integrity violation. **The minimum penalty for cheating on a programming assignment is a grade of -100 on the assignment.** If you are tempted to copy because you're running late, or don't know what you're doing, you will be better off missing the assignment and taking a zero. Providing your code to someone is cheating, just as much as copying someone else's work.

DO NOT copy code from the Internet, or use programs found online or in textbooks as a "starting point" for your code. Your job is to design and write this program from scratch, on your own. Evidence of using external code from any source will be investigated as a potential academic integrity violation.

For this assignment, you will write a program that draws your school's letters on PennSim's graphic display in the appropriate color. The program user will be able to change the color of the letters.

Program Specification

The program must start at address x3000.

The program will draw your school's letters (NCSU, UNCA, UNCW) onto the center of the graphics screen. The initial color shall be the main color for your school (i.e. N.C. State Red, etc.). These letters shall be centered in the screen (L/R, Top/Bottom) and shall be 84 pixels tall. The width and style is your choice.

The PennSim graphics display is the same as described in Program 2.

The user interacts with the program using one-character commands. The commands are typed on the keyboard, but are not printed to the console display. Nothing will be printed to the console during the execution of this program. The program will wait for a keystroke, perform the corresponding command, and repeat. If the keystroke does not correspond to a legal command, it will have no effect.

There are five commands for changing the letters color:

Command Character	Action
r	Change color to *Red*.
g	Change color to *Green*.
b	Change color to *Blue*.
y	Change color to *Yellow*.
space	Change color to *White*.

q	*Quit.* The program shall display a newline and message "Thank you for playing", then HALT.

NOTE: You **MUST** use/load **p3os.obj** for this code to run properly!

Details

The PennSim graphics display is bit-mapped, meaning that each pixel has a corresponding memory location. The specifications are the same as in Program 2.

Pixel Color

As mentioned above, the value of the pixel address determines the color of the pixel. The 16 bits contain 5 bits for each RGB component of color: bits [14:10] for red, [9:5] for green, and [4:0] for blue. Bit 15 is ignored. The higher value of a component, the more of that color is present. The table below gives the color values (in hex) needed for this program.

Color	Value
Red	x7C00
Green	x03E0
Blue	x001F
Yellow	x7FED
White	x7FFF
Black	x0000

Miscellaneous

- You **MUST** use at least **six (6) subroutines** for this program! More is ok.
- For more explanation about the PennSim display, see the PennSim Reference Manual.

Hints and Suggestions

- As always, **design before you code**! Draw a flowchart to organize and document your thinking before you start writing the program.
- Make a large spreadsheet 124 x 128 cells to plan your display.
- **Work incrementally!** For example, implement one command at a time. Make sure the program works before moving on to the next command. This way, you always have working code.
- It's not a bad idea to submit each working version of your program to Moodle. Then, if your machine crashes (it happens!), you haven't lost everything. Each time you submit, it overwrites the previous submission, so you can submit as many times as you like. But don't expect that we can

recover some previous version of your code if you accidentally clobber it. (You should have some sort of backup system for your schoolwork, right?)

- *Test your program with a wide variety of inputs.* Make sure that you have tested the "corner cases," such as reaching the border of the display.
- Use the PennSim simulator and assembler. There are other simulators and assemblers out there, but there are some differences. Your program will be graded using PennSim, and no other tools will be used or considered.
- You must use at least six (6) subroutines for this program.

Administrative Info

Any corrections or clarifications to this program spec will be posted on the **Discussion Forum**. It is important that you read these postings, so that your program will match the updated specification. (I recommend strongly that you subscribe to the forum, so that you will not miss any updates or corrections.)

What to turn in:

- Assembly Language Source file – it must be named **school.asm**. Submit via **Moodle** to the Program 2 assignment.
- DO NOT submit .obj or .sym files. Do not submit a .txt file, a .doc file, or anything like that. It must be a simple text file with the proper .asm extension. If we are unable to open or interpret your file, you will get a zero for the assignment (even if it has the right name!).

Grading criteria:

5 points: Submission of any assembly code file by due date.

5 points: Correct type of file, submitted with the **proper name**. (No partial credit!! These are essentially FREE POINTS! Don't screw it up.)

10 points: **Program is complete** and **assembles with no warnings and no errors** using the PennSim assembler. To be "complete," there must be code that makes a reasonable attempt to meet the program specs. Programs that do not assemble will not be graded any further. (For warnings, points will be deducted, but the program will be tested for correctness.)

10 points: **Proper coding style, comments, and header**. Use indentation to easily distinguish labels from opcodes. Leave whitespace between sections of code. Include *useful* comments and *meaningful* labels to make your code more readable. Your file **must** include a header (comments) that includes your name and a description of the program. Don't cut-and-paste the description from the program spec – that's plagiarism. Describe the program in your own words. This category is somewhat subjective, but the goal is that your program should be easy to read and to understand.

70 points: The program **handles all commands correctly**:

(30 points) Displays your school letters in the correct initial color. You are free to alter the color numbers to reach an intermediate shade as you wish.
(25 points) Change colors (5 pts each)
(10 points) Use at least 6 subroutines
(5 points) Quit.