SUMMER 2013 CSE 2421 LAB1

*** Assigned: Wednesday June 12th ***

*** Due Date: Monday June 17th by 11:59PM ***

Objectives:
- Standard I/O
- Arithmetic statements
- IF structures
- Looping structures

REMINDERS:
- You are allowed to work in pairs for this assignment.
- Every lab requires a README file. This file should include the following:
  - Student name(s) - to avoid the 10% deduction (as explained in the course syllabus) if working with a partner
  - Effort distribution for each contributor (assumed 100% if you are not working with a partner)
  - Total amount of time to complete the entire lab
  - Short description of any concerns, interesting problems or discoveries encountered, or comments in general about the contents of the lab
  - Directions on how to run the solution(s), as needed, as long as they don’t conflict with the information given in the lab assignment.
- You should aim to always hand an assignment in on time. If you are late (even by a minute – or heaven forbid, less than a minute late), you will receive 50% of your earned points for the designated grade as long as the assignment is submitted by 11:59pm the following day, based on the due date listed on the syllabus and confirmed by the instructor. If you are more than 24 hours late, you will receive a zero for the assignment and your assignment will not be graded at all.
- Any lab submitted that does not compile and run WILL RECEIVE AN AUTOMATIC GRADE OF ZERO. No exceptions will be made for this rule - to achieve even a single point on a lab, your code must minimally compile and execute without crashing.
- You are welcome to do more than what is required by the assignment as long as it is clear what you are doing and it does not interfere with the mandatory requirements.
- You are responsible for making sure that your lab submits correctly.
- Be sure that you are aware of the Coding Style file information posted on my “Day by Day” website. The issues that you should keep in mind when writing your programs are readability and maintainability. Thus, things like indenting, descriptive variable names, and plenty of comments are very important. If you have any questions on coding style issues, please ask.

GRADING CRITERIA (approximate percentages listed)
- (20%) The code and algorithm are well commented.
  - A code block should be included in the main program including the programmer name(s) as well as explaining the nature of the problem and an overall method of the solution (what you are doing, not how).
  - A short comment should be included for each logical or syntactic block of statements
- (20%) The program should be appropriate to the assignment, well-structured and easy to understand without complicated and confusing flow of control.
- (60%) The results are correct, verifiable, and well-formatted. The program correctly performs as assigned.

LAB DESCRIPTION - Mandatory filename → lab1.c
You are creating a character for a video game and playing against the computer generated character to practice your gaming skills. Each character, yours and the computer’s, has the same 5 skills designated by the letters A-E. You get to choose the 5 abilities that each character is gifted with. These talents can be weapon based, super-hero quality options, or any other traits you can think of to play and win a contest between these two characters. Have fun with it. Be creative.
Name your character (and even the video game if you like although optional) then determine the “skills” and specify how each skill beats another skill – for instance, “laser melts (or burns through) steel”.

**RULES OF THE GAME**

| A beats B | A beats D |
| B beats C | D beats B |
| C beats D | B beats E |
| D beats E | E beats C |
| E beats A | C beats A |

**THE RESULT** (once a valid input is received)

- Output the randomly generated computer character skill letter chosen with a descriptive message.
- Determine the winner remembering that a tie is possible.

**Your program should:**

1. Prompt the game player for one of the 5 options.
   - Output a list of all five skills to the screen by designating each skill with a letter A-E but also accept a-e (i.e. lower case or upper case).
   - Ask the user to enter a letter associated with a skill. Be sure to give the user a non-ambiguous message about how to enter the input information.
   - HINT: there are built-in functions in the C standard library that convert upper case to lower case; and lower case to upper case.
   - You cannot assume that the user will enter a correct value, so be sure to check that a valid input was received. At this point, the user may decide to stop, so be sure to ask the user if they want to enter another skill or would they like to quit. If you prefer, you can have a “quit” option along with the skills listed (your choice how to implement this).

2. Randomly determine the computer player option.
   - How do you obtain a random number for 5 options? There is a random number generator function (which you will need to look up). The number returned from the function is rather large, though. How do you use this large random number to generate one of three options? HINT think about what the modulus operator does. Also, be careful to choose a function that is as random as possible as you don’t want the computer choosing the same skill each time – you might even want to add a little math to help things along.

3. Declare a winner.
   - Be sure to output the randomly generated computer character skill letter chosen with a descriptive message.
   - Based on the rules of the game defined above, determine a winner and output a congratulatory message to the monitor announcing the winner; remembering that a tie is a viable outcome.

4. Repeat the game until the user declines to play again.
   - Be sure that you play the game at least once i.e. do not prompt the user asking if they want to play again before they play the first time. You can assume that since the player is executing your program that they want to play the game at least once.

**LAB SUBMISSION**

You should submit all your lab assignments electronically using the submit command. The format of submit command is as follows:

```
submit c2421ad labname files-to-submit
```

where, the c2421xx is different per course section (if multiple ones exist), labname is the lab you are working on (lab1, lab2, etc.) and files-to-submit is a list of the file(s) that make up the lab. For more information about the submit command, review this section in your pre-lab practice problem.

The labname for this lab is: **lab1**
Be sure to submit the following files: lab1.c lab1README