Cpt S 121 - Program Design and Development

Programming Assignment 7: Educational Math Program

Assigned: Monday, April 7, 2014
Due: Wednesday, April 23, 2014 by midnight

I. Learner Objectives:

At the conclusion of this programming assignment, participants should be able to:

- Apply and implement all your problems solving and C skills developed this semester!
- Apply and implement pointers in C
- Pass output parameters to functions
- Manipulate and split strings

II. Prerequisites:

Before starting this programming assignment, participants should be able to:

- Analyze a basic set of requirements and apply top-down design principles for a problem
- Apply repetition structures within an algorithm
- Construct while (), for (), or do-while () loops in C
- Compose C programs consisting of sequential, conditional, and iterative statements
- Eliminate redundancy within a program by applying loops and functions
- Create structure charts for a given problem
- Open and close files
- Read, write to, and update files
- Apply standard library functions: fopen (), fclose (), fscanf (), and fprintf ()
- Compose decision statements ("if" conditional statements)
- Create and utilize compound conditions

III. Overview & Requirements:

Mathematics is one of the most important, yet most difficult, subjects to teach and learn. We have all heard of the saying "Practice makes perfect".
Well I'm a huge believer in this saying and would like a software program which can generate various arithmetic problems and evaluate answers supplied
by the user.

For this assignment you will need to design and construct a basic math program which targets elementary school children. Your program must adhere to the following steps and requirements:

1. (5 pts) A user interface with options similar to the following:
    a. Learn about how to use the program
    b. Enter your initials (3 individual characters...)
    c. Difficulty selection
    d. Start a new sequence of problems
    e. Save and Quit
2. (25 pts - 5 pt/level) Generate mathematical problems based on the difficulty level selected. You must implement the following five levels of difficulty:

a. Level 1 - Easy: Includes addition and subtraction problems, with positive single digit operands and up to three terms only (i.e. d1 + d2 - d3 =)

b. Level 2 - Fair: Includes multiplication problems, with positive single digit operands and up to two terms only (i.e. d1 x d2 =)

c. Level 3 - Intermediate: Includes division problems, with positive single digit operands and up to two terms only (i.e. d1 / d2 =);

Note: results should be shown in the form Num R Remainder, i.e. if the problem is 5 / 3, then the answer should be provided as 1 R 2.

d. Level 4 - Hard: Includes a mix of addition, subtraction, multiplication, and division problems, with positive and negative single digit operands

and up to three terms only (i.e. d1 + -d2 / d3 =); Hint: you may have to first find a common denominator.

e. Level 5 - Impossible: Includes a mix of addition, subtraction, multiplication, and division problems, with positive and negative

two and three digit operands and up to four terms only (i.e. dd1 + -ddd2 x ddd3 / dd4 =); Hint: you may have to first find a common denominator.

3. (5 pts) Allow the user to enter an answer corresponding to a generated math problem.
4. (10 pts) Evaluate the answer provided by the user. The user gets a certain number of points for correct answers

and loses points for incorrect answers. The number of points should directly relate to the difficulty of the problem.

5. (10 pts) Each level must generate a sequence of ten problems
6. (10 pts) Within each level, problems should become a little more difficult as the user enters correct answers
7. (10 pts) Once the user quits the program, output the user's initials and total score to a file

Your program must implement error checking where appropriate. It also must use strings, pointers, and output parameters where appropriate.

BONUS Opportunities:

1. Implement a "load previous progress" feature (up to 10 pts)
2. Implement a "help" feature, which illustrates step-by-step how to solve various addition, subtraction, multiplication, and division problems (up to 15 pts)
3. Others? (up to 15 pts)

IV. Submitting Assignments:

1. Using the Angel tool https://lms.wsu.edu submit your assignment to your TA. You will "drop" your solution into the provided "Homework Submissions" Drop Box under the "Lessons" tab. You must upload your solutions as <your last name>_pa7.zip by the due date and time.
2. Your .zip file should contain two project workspaces. One for each project above. Within each project you must have at least one header file (a .h file), two C source files (which must be .c files), and project workspace. Delete the debug folder before you zip the project folder.
3. Your project must build properly. The most points an assignment can receive if it does not build properly is 65 out of 100.

V. Grading Guidelines:

This assignment is worth 100 points. Your assignment will be evaluated based on a successful compilation and adherence to the program requirements. We will grade according to the following criteria:

- 75 pts for adherence to the instructions stated above (see the individual points above)
- 15 pts for appropriate usage of pointers and output parameters
- 10 pts for appropriate top-down design of functions and good style
- Up to 40 bonus pts for various added feature