Assignment #3

Introduction to C Programming – COP 3223

Objectives

1. To reinforce the use of loops.
2. To learn how to read information from files.

Introduction: Exercise UCF

UCF is committed to student health and already has a great Recreation and Wellness center on campus for students to use. To encourage students to find out more about themselves and their fitness, they are implementing some new initiatives.

Exercise UCF is one of their new programs to educate students about weight, calories, and Body Mass Index. UCF would like to the help of the Computer Science division to create some of these programs.

Problem: Strength Training Workouts (strength.c)

A balanced workout regimen contains cardio workouts, strength training, and flexibility training. The participants of Exercise UCF have been diligently recording their workouts since they began this new program. Now Exercise UCF would like to have an automatic program that will process the participants' data on their strength training workouts.

The participants must record the number of weeks they have been participating, the number of workouts they completed each week, the number of sets they completed in each workout, and the number of repetitions (reps) that each set contained.

A "good" workout should have about 3 sets of ten or more reps for each major muscle group being focused on. Most new workout regimens recommend focusing on a major group once a week: one workout given to upper body muscle groups, one workout given to lower body muscle groups, and one given to core muscles. Alternatively, some people will do two workouts of full-body strength training a week and one abs workout.

To process the information given, your program should store the name of the participant and the number of weeks that they have data for. Then, for each week you will need to store the number of workouts. For each workout, you should consider the number of sets that they completed. Check the number of reps for each set – only sets that contain 10 or more reps should be counted.

You should print the information for each week in a small visual chart. Please check the output specification for more information.

At the end of the program, you should print out the participant's best workout by indicating which week it occurred in. This should be determined by the workout with the highest number of sets that count. Please check the output specification for more information.

Input Specification
1. The input will be in a file called input.txt

Input File Format
The first line of the file will contain the participant's name.

The second line of the file will contain a single positive integer, *n(1< n < 10)*, representing the number of weeks that the participant has data for.

The first line of each week will contain a single positive integer, *w(1 < w < 10)*, representing the number of workouts for that week. The next line will contain a single positive integer, *s(1 < s < 10)*, representing the number of sets for first workout. The next *s* lines will each contain a single positive integer, *r (1 < r < 50)*, representing the number of repetitions in each set.

You should keep track of the workout with the highest number of sets that count. Counting sets have 10 or more reps. For the workout with the highest number of counting sets, you should also store the average number of reps for sets in that workout.

A sample input file is below in the sample run.

Output Specification
Output to the screen. The first line should contain the participant's name. Following should be a small chart that shows each week the user has been participating, with each workout completed in that week and a * for each counting set completed in that workout.

Name
Week 1
Workout 1: ****
Workout 2: **
Workout 3: ***

Week 2
...

Your best workout was in week X and contained Y sets of an average of Z reps.

Output Sample
Below is a sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity's sake.)

Sample Input File (input.txt)

```
Karla
2
3
3
10
12
6
2
10
15
4
10
10
10
10
1
2
15
15
```

Sample Output

Karla
Week 1
Workout 1: **
Workout 2: **
Workout 3: ****

Week 2
Workout 1: **

Your best workout was in week 1 and contained 4 sets of an average of 10 reps.

Deliverables

One source files – *strength.c* – is to be submitted over WebCourses.

Restrictions

Although you may use other compilers, your program must compile and run using Code::Blocks. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

Grading Details

Your programs will be graded upon the following criteria:

1) Your correctness

2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.

3) Compatibility – You must submit C source files that can be compiled and executed in a standard C Development Environment. If your program does not compile, you will get a sizable deduction from your grade.