**Assignment #2**

**Introduction to C Programming – COP 3223**

**Objectives**

1. To give students practice at typing in, compiling and running simple programs.

2. To reinforce knowledge of assignment statements and arithmetic expressions

**Introduction: Ultimate Computing Fun**

A new game store is opening in your area! Ultimate Computing Fun (UCF) Games is looking for a home for their wide variety of computer games. In order to get started, they’re going to need to research what kinds of games people want to see on their shelves, get some cash registers, and stock the shelves of their new building. UCF Games has decided to make you a part of their growing business.

**Problem: Stocking the Shelves (shelf.c)**

Now that UCF Games has received its first shipment, it’s time to stock the shelves! Like any new store, the shelf space is somewhat limited. For this problem, you will design a program that will calculate how many games will fit on each shelf.

The user will enter the length of the shelf (in inches) and the length of each game case (in inches). You must first determine how many games can fit on each shelf.

The user will also enter the number of shelves that are present and the number of games they want to display. You must determine how many of the games will be displayed and how many must be kept in the back room.

**Input Specification**

1. The length of the shelf will be a positive integer less than 100.

2. The length of the game case will be a positive integer less than 10.

3. The number of shelves will be a positive integer less than 50.

4. The number of games to be displayed will be a positive integer less than 500.

**Output Specification**

Output the number of games that can fit on a shelf using the following format.

X games can be displayed on 1 shelf.

Output the other results using the following format.

X games can be displayed on Y shelves. Z games will have to be stored in the back.

**Output Sample**

Below are some sample outputs of running the program. **Note that these samples are 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 Run #1**

**What is the length of the shelf, in inches?**

*72*

**What is the length of the game, in inches?**

*6*

**12 games can be displayed on 1 shelf.**

**How many shelves are there?**

*5*

**How many games do you want to display?**

*200*

**60 games can be displayed on 5 shelves. 140 games will have to be stored in the back.**

**Sample Run #2**

**What is the length of the shelf, in inches?**

*84*

**What is the length of the game, in inches?**

*5*

**16 games can be displayed on 1 shelf.**

**How many shelves are there?**

*8*

**How many games do you want to display?**

*200*

**128 games can be displayed on 8 shelves. 72 games will have to be stored in the back.**

**Deliverables**

One source files – *shelf.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.