CSCI 1312 (Introduction to Programming for Engineering), Fall 2015

Homework 3

Credit: 20 points.

1 Reading

Be sure you have read (or at least skimmed) the assigned readings from chapter 5.

2 Programming Problems

Do the following programming problems. You will end up with at least one code file per problem. Submit your program source (and any other needed files) by sending mail to `bmassing@cs.trinity.edu`, with each file as an attachment. Please use a subject line that mentions the course and the assignment (e.g., "csci 1312 homework 3" or "CS1 hw3"). You can develop your programs on any system that provides the needed functionality, but I will test them on one of the department's Linux machines, so you should probably make sure they work in that environment before turning them in.

Some words about checking for non-numeric input: Doing a great job of "validating input" is probably outside the scope of this course — it turns out to be not an easy problem to solve in C — so in the programs you write for homework, generally speaking what I have in mind is for you to just check the return value from `scanf`, as I do in the examples in class, and bail out of the program if it indicates an error. You may be tempted to make your program prompt the user again, but don't do that either — it breaks my semi-automated testing and again is not trivial to do in C. Just checking the return value from `scanf` is far from perfect but at least ensures that the variable(s) read from standard input have sensible values, which is good enough for our purposes right now.

1. (10 points) Write a C program that asks the user for three integers and prints them in order from smallest to largest, or an error message if what was entered is something other than three integers. (This will turn out to be a special case of a more general problem of putting things in order. Don't try right now to solve the general problem; just solve it for three numbers using what we've discussed in class so far up through 9/23.) *Clarification:* Notice that the numbers do not have to be distinct — for example, the input could be three values all the same, or two the same and one different, or all three different.

2. (10 points) For this problem, imagine a company (call it Acme Caps) that makes and sells custom baseball caps, priced as follows:

   - The minimum order is 50 caps, for which the price is $100. Acme will fill orders for fewer caps, but will still charge $100 for them.
   - For more than 50 caps, Acme charges the minimum of $100 for the first 50 caps, plus $5 for each additional group of 5 caps (rounding up if the number is not a multiple of 5). (So for example 55 caps cost $101, and for 51 caps the price is the same.)

- Orders of more than 200 caps get a volume discount; Acme charges its minimum ($100) for the first 50 caps, $5 per group of 5 for the next 150 caps, and then only $4 for each group of 5 caps beyond the 200-cap threshold. (So for example 205 caps cost $254, and for 201 caps the price is the same.)

(Is this a realistic business model? I don't know! but I think it makes a reasonable programming problem for this assignment.)

Write a C program that prompts the user for the number of caps and prints the corresponding price. Of course(?), the program should print an error message and stop if input is not numeric or numeric but less than 0. The program should also print an informational message if the price would be the same for a larger number of caps (i.e., the number entered is less than the "minimum" order, or not a multiple of 5). It should also print a message if the order qualifies for the volume discount.