CS Department Challenge #2!!!!

This timed challenge has three programming problems.

Success will require that you read the problems and think about program structure and design before coding.

A good design will result in an efficient program and easy coding in all these problems.

Scoring will be based on both having a correct program *and* being first.

I hope you enjoy the meta-story around them.

1) It was a dark and stormy night and as you unlock your apartment door a huge hand grabs you by the shoulder. Turning around, there is a huge, bearded man in a large coat. He smells of stoats or perhaps aged skunk. In any case, something mammalian and unpleasant.
   Before you can say anything, he says, “Harry, you’re a wizard.”
   “My name’s not Harry”
   “Oh yeah, but you’re a mathematician ain’t yah?”
   “A computer scientist.”
   “Close enough to a wizard for us.”
   He touches his pink frilly umbrella to an autographed manuscript copy of Kernigan and Ritchie’s “the C programming language, 1st edition”, and you are transported to a drafty castle in Scotland.

   Yet another bearded bloke meets you there. “Harry, we have a problem.”
   “My name’s not Harry.”
   It appears that unless they balance their books, the late Lord Voldemort’s cousin Billy will foreclose on the castle. Since the beer is weak, watery, and tastes of butter and the only edible thing in the school cafeteria is the vegan haggis, you agree to help.

   The problem is that the wizarding money system is to say the least eccentric. 13 knuts to a sickle and 17 sickles to a galleon.

   Your task is to write a program that adds a column of numbers in this system and comes up with the correct answer. Some of the numbers will be negative. If the leading digit is negative then all the digits are negative (e.g. -1g,3s,8k is -1 -3 -8)

   Sum:

Galleon	Sickle	Knut
5	11	5
3	1	8
25	0	7

-10	16	12

A hint: the naïve way of solving this problem, namely working in digits in an oddball radix is not the best solution.

2) Your hosts agree to return you home so that you can rejoin the Hackathon. In their whimsical way, they have an accident with a time turner and an exceedingly bad sense of direction. Instead of home, you find yourself in Regency Steventon, England, outside of Jane Austin's house.

   This is not exactly a tragedy as Miss Austen and her family are good company with elegant manners. The beer is real ale, the sherry a true madeira, and the brandy run. However, on the afternoon before the Steventon assembly, where they have invited you to the dance, you interrupt Miss Austen crying in her study.

   She must balance the household accounts, in pounds, shillings, and pence. In addition, her publisher, Edgerton, seems to have been cheating her. So it will be necessary to calculate the percentage of a number.

   Your task is to write a program that does these things.

   The pounds, shillings, and pence system is 12 pence = 1 shilling, and 20 shillings = 1 pound. We can skip coins like crowns (5 shillings), but have to handle the smallest coin, a farthing which is ¼ penny.

   So 1L, 15s, 1 ¾d is a perfectly normal amount of money. A negative sign in front of the entry indicates that all the entries are negative.

   Sum:
   1L 19s 5d
   -2L 5s 1 ½ d
   10L 8s 5 ¼ d

   Calculate 25% of 435L 8s 5d

   Hint: if you've structured your first program correctly, you just have to change a few constants to write this code.

3) On the way back from the Steventon assembly you run into a small party of aliens. They've been having a fun time making crop circles and are ready to return to their home planet. After you chat with them, they are willing to return you to your correct time and place.
Unfortunately, as you are zooming past Pluto's orbit, they tell you that their trig tables are missing. Finding their way, and more importantly your way, home without these is to say the least, problematic.

You offer to supply the missing data from your laptop, using the Python math functions.

However, there are two big hitches with this.
a) They don't use floating point numbers.
b) They work in octal. (Having only three fingers and a thumb).

Their computer uses rational numbers instead of floating point. Non-integer numbers are represented as a fraction of two integers. (e.g. 0.5 is ½ 0.25 is ¼, 0.3333333 is 1/3 …) The integers used in the fractions are limited to 16 bits. (a maximum of 65535 or octal 177777)

Because of symmetry and trig identities you need only calculate the first 45 degrees (or octal 55 degrees) and the sin function.

You will write a program that finds (0-65535)/(0-65535) representations of the decimal value of sin. Output the octal representations of:

Angle, numerator, denominator

For all 46 (python 0o56) degrees from zero to 45 in one-degree increments.

Hint1: in python: oct(an_integer) returns the octal string.

Hint 2: the naïve approach of using two nested loops to find a good approximation will be too slow. A better approach is to use Fairey fractions.

Fairey fractions

$$\frac{a}{b} < \frac{(a+c)}{(b+d)} < \frac{c}{d}$$

If you start from a pair of fractions that bound the desired value and construct a series of Fairey fractions, you can tighten the bound. Just remember to stop when a+c or b+d gets too big.

Email your correct answers to rwh@gsu.edu