Prog2: Floating bananas (CS220-02, 05)

Develop a C (or C++) program to let you pay and receive change for purchasing bananas by the pound. The purpose of this programming assignment is NOT to challenge your ability to program, but to use your successfully running program to find simple instances where the act of calculating what you owe and what your change should be is not quite accurate even though the numbers involved are not extraordinary… not extraordinary for a computer, anyway.

Your program must read in 3 float values (you MUST use float… not double or anything else):

- pounds: real-valued pounds of bananas to purchase
- price: price per pound for bananas
- cash: how much money you'll be paying with

Your program must calculate a few float values (again, the float type is required here):

- total: this is the pounds of bananas multiplied by the price per pound
- change: this is the amount of cash paid minus the total

If the user tries to underpay, they should be creatively punished (extra credit described below).

Wow! What a challenge!!! It's not meant to be tricky or anything. This assignment is just meant to let you write some obvious code that surprisingly doesn't work all the time.

Example runs should work like the following example from my own solution (blue text is input by user)…

```
How many pounds of bananas (eg, 1.7): 6
What is the price per pound (eg, $0.59): $0.89
Your total is $5.34
How much cash will you pay with (eg, $4.43): $30000
your change is $29994.7                    ←←←←←← Ooooh!  I just got richer!
```

```
How many pounds of bananas (eg, 1.7): 2.05
What is the price per pound (eg, $0.59): $2.99
Your total is $6.1295
How much cash will you pay with (eg, $4.43): $5
Tryin' to cheat me?... Back of the line, chump!
```

SEE NEXT PAGE FOR FURTHER DETAILS

To complete this assignment, you will need to upload only your single C or C++ source file to Brightspace **AND** in a BIG comment at the top of your source (eg, the file-level comment) list 4 sets of input you find through trial and error that produce incorrect results. For example, I could write, "6 0.89 30000" in the comment to represent the inputs that lead to the incorrect result in my example above.

EXTRA CREDIT

For a ridiculous 20% possible extra credit, using only standard C/C++ code, do something creative / zany for the circumstance the user tries to underpay. The example I gave above, "Tryin' to cheat me?... Back of the line, chump!" would not earn any extra credit. Maybe if there was an incredibly fabulous play on words or a great dad-pun it would earn at least some portion of the extra credit. I can't be concrete or objective here about what I'm looking for… this will be wildly subjective, but the more I'm impressed, the more extra credit I will award. Maybe loops, pseudo-animations, or some other fantastical construction… I really can't be more specific or give examples or that will be all anyone bothers to do.