Assignment SPIM 2

Due Date: April 3

Purpose
Finally, more SPIM! More fun than NCAA brackets! In this problem, you will manipulate floats and characters, and write a couple of functions (procedures). You may have done something similar in Comp 115, so how hard could it be?

Problem
Smells Fargo has hired you to write some of their financial tools (I hope they know what they’re doing!). You will write a module for their web pages that will help customers calculate the cost of their mortgages. Given an initial loan amount and other data, you will calculate the monthly payment and the total cost of the mortgage.

Input
You may assume that all input will be correct. Actual data would be gathered through a web interface. You will simulate this data gathering through the usual prompts, etc. No error checking is required.

The program should prompt for the initial loan amount (a float), the number of years (an integer), and the interest rate percentage (a float).

Note that the interest rate will be input as a percentage; that is, if the rate is 3.9%, then the user will input 3.9, and not 0.039.

After the results are displayed, the program should ask if the user would like to repeat the process. A one-letter input of “y” or “Y” indicates the program should loop; any other letter means the program should stop.

Output
The program should first display all of the input information in a nicely formatted way. It should then display:

- the monthly payment
- the annual payment amount
- the total interest paid
- the total of all payments

Be sure to label all values.

All output should be in a clearly readable format, aligned as best you can. You have seen how awful floating points are represented and displayed. Therefore, all monetary values must be rounded to the nearest penny. This is a major part of the program. Finally, when the program ends, a “Thank you” or other suitable message should be displayed.
Specifics

- Your program should include at least two functions (procedures): One should get the user input data and pass the results back. In the real product, this would be replaced by the web page interface. Note that this is a bit of a problem because there are only two registers for returning values. The second procedure should be passed a computed result and display that value rounded to the nearest penny. Feel free to add additional functions or procedures.

- All dollar values should be displayed with only two significant decimal places. No extra zeros should be displayed.

- Rounding to the nearest penny is non-trivial. One way to do this is to use the instruction to convert to integer to truncate extra decimal places. **Note that floats converted to integers must still be stored in floating point registers.** This is annoying, but you can store an integer that is currently in a floating point register in memory as a word, using the `s.s` instruction; alternatively, you can use the `mfc1` instruction to move a value into an integer (word) register. Storing the value in one of these ways will ensure that extra decimal places are indeed truncated. Then the trick is to display the final dollar value as integers rather than a floating point value so that you will see only two decimal places.

- Follow the style conventions as in the first assignment. Comment all functions (procedures), including the parameters.

Notes

- Once again, the general problem is fairly trivial. But the rounding to two decimal places is not straightforward in SPIM, and will take some time to do properly. The character input is also a bit annoying, but we’ve done something similar in lab. Don’t wait too long to start!

- You can look up how to compute mortgages/interest via The Google. Learning about mortgages is a good life-lesson, either from an investment perspective or, at least in the United States, from a debt perspective.

- Turn in a printed copy at the start of class on April 4th.

- Turn in your source code to me via email using the naming convention as follows: `firstInitial,lastNameSPIM2.a`, as in `mgousieSPIM2.a`.

---

The computer is no better than its program.
– Elting Elmore Morison, in *Men, Machines and Modern Times* [1966]

And this xspim program is pretty bad.