

Assignment MIPS 2

Due Date: April 7

Purpose

Finally, more MIPS! More fun than the start of baseball season (well, if you're not a Red Sox fan, anyway)! In this problem, you will manipulate integers and floats, and write a couple of functions (procedures). A major portion of the program will be to convert floating point values to integer and vice-versa, and to display the results in the correct format.

Problem

Smells Fargo has hired you to write some of their internal financial tools (*big bucks for you!*). You will write a module for their web pages that will help loan specialists calculate the true cost of mortgages. Given an initial loan amount and other data, you will calculate monthly payments and the total costs of mortgages.

Input

You may assume that all input will be correct. If this were part of the actual project, the 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 double), the number of years (an integer), and the interest rate percentage (a double).

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.

Output

The program should first display all of the input information in a nicely formatted way. It should then display the following data in the proper format:

- the monthly payment as a double, a real number **rounded** to two decimal places (that is, rounded to the nearest penny), and a real number **truncated** to two decimal places.
- the total of all payments as a sum of the above doubles, the total as a sum of the rounded values, and the total as a sum of the truncated values.
- the total interest paid, again as double, rounded, and truncated values.

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 that are rounded or truncated must be shown with two decimal places (even if they are 0). This is a major part of the program.

Sample Output

For example, suppose the principal (loan amount) is \$100000, the interest rate is 7.25%, and the length of the loan is 20 years. The output should look similar to the following:

```
Loan amount      : $100000.00
Interest rate    : 7.25%
Loan term        : 20 years
```

Exact Values

```
Monthly Payment: 790.37598492
Total Payments  : 189690.23638024
Total Interest  : 89690.23638024
```

Rounded Values

```
Monthly Payment: 790.38
Total Payments  : 189690.24
Total Interest  : 89690.24
```

Truncated Values

```
Monthly Payment: 790.37
Total Payments  : 189690.23
Total Interest  : 89690.23
```

Note that the monthly payments need not add up to the total payments exactly. This is because the monthly payment is not exact. If you were to take out an actual mortgage, you would find one payment/year is not the same as the rest to take into account the fractional cents over multiple months.

Specifics

- Your program should include two functions (procedures):
 1. A function that is passed x and y , where x is a double precision floating point value and y is an integer. The function should return a double, the result of x^y .
 2. A function that is passed a double precision floating point value and returns that value rounded to the nearest penny (hundredths decimal place). Use conventional rounding (.5 rounds up).

Additional function(s) are encouraged but not required.

- All of your exact calculations should be done using double precision floating point values.
- The exact values should be displayed with whatever number of decimal digits MARS will display; no need to format the output.
- All other dollar values should be displayed with exactly two significant decimal places. No extra zeros should be displayed; conversely, a value such as 3.4 should be displayed as 3.40. This may be a little tricky! There are no formatting instructions in MIPS.

- Arrange the output as neatly as possible, as aligned as possible (see above).
- 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 MIPS, and may take some time to do properly. Similarly, just displaying to two decimal places can be non-trivial. Do not wait to start!
- You can look up how to compute mortgages/interest via The Google. Look up “mortgage formula” for how to do the calculations and “mortgage calculator” to check your answers. Learning about mortgages is a good life-lesson, either from an investment or debt perspective.
- Turn in your source code to me via Canvas using the naming convention as before: first initial + last name + MIPS2.s|asm (the “|” means “or”), as in mgousieMIPS2.s.
- Turn in a printed copy at the start of class on April 8th. Don’t forget to write and sign the Honor Code.

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