

Lab 2

This lab explores control flow and basic computations. An extension of this lab, to implement a bisection search version of Part B, will be part of the problem set.

Part 0: Basic test

The goal of Part 0 is to get used to using git and GitHub to update your code and [Travis CI](#) to monitor whether or not you are passing the unit tests. For this part, update the function so that it behaves as described in its docstring, then verify that you are passing the test for this part on TravisCI.

Note: it's possible to run the tests locally and not rely on online services. To do this, from a terminal, you will need to run `pytest -v test.py`.

Part A: Minimum payments

The goal of this problem is to compute the number of months that would be required to pay off a credit card balance, given some starting balance, an APR (annual percentage rate) and a monthly payment equal to the loan's minimum monthly payment.

The program should ask the user to input:

1. The starting balance (in dollars)
2. The APR (as a percentage)
3. The minimum payment (in dollars)

Note that the effective periodic rate can be computed from the APR using the formula:

$$effective_rate = (1 + APR)^{(1/n_periods)} - 1$$

In the case of an APR, this means that the monthly rate is given by:

$$effective_rate = (1 + APR)^{1/12} - 1$$

Hint: as a rough outline, your code will need to carry out the following steps:

1. Retrieve user input for the 3 input variables.
2. Initialize some state variables

3. Loop a month at a time, computing additional interest added to the balance and the amount deducted by that month's payment.

Here is an example test case to check your code (**your program should print results in this format**):

```
Enter the starting balance: 5000
Enter the APR: 18
Enter the minimum payment: 100
Months to pay off: 86
```

(note that the final line is from a print statement, not user input!)

Part B: Pay off in one year, simple

In part A, you computed how long it would take to pay off a loan given some balance, APR, and monthly minimum payment. What about if we wanted to know what fixed amount we would need to pay, per month, to pay off the loan in one year?

As a constraint on this problem, you should only consider possible monthly payments that are multiples of \$10.

Here is an example test case to check your code (**your program should print results in this format**):

```
Enter the starting balance: 5000
Enter the APR: 18
Monthly Payment: 460
```