PROBLEM 2: PAYING DEBT OFF IN A YEAR (15/15 points)

Now write a program that calculates the minimum **fixed** monthly payment needed in order pay off a credit card balance within 12 months. By a fixed monthly payment, we mean a single number which does not change each month, but instead is a constant amount that will be paid each month.

In this problem, we will *not* be dealing with a minimum monthly payment rate.

The following variables contain values as described below:

1. `balance` - the outstanding balance on the credit card
2. `annualInterestRate` - annual interest rate as a decimal

Help

The program should print out one line: the lowest monthly payment that will pay off all debt in under 1 year, for example:

```
Lowest Payment: 180
```

Assume that the interest is compounded monthly according to the balance at the end of the month (after the payment for that month is made). The monthly payment must be a multiple of $10 and is the same for all months. Notice that it is possible for the balance to become negative using this payment scheme, which is okay. A summary of the required math is

found below:

Monthly interest rate = (Annual interest rate) / 12.0
Monthly unpaid balance = (Previous balance) - (Minimum fixed monthly payment)
Updated balance each month = (Monthly unpaid balance) + (Monthly interest rate x Monthly unpaid balance)

Test Cases to Test Your Code With. Be sure to test these on your own machine - and that you get the same output! - before running your code on this webpage!

Click to See Problem 2 Test Cases

The code you paste into the following box **should not** specify the values for the variables `balance` or `annualInterestRate` - our test code will define those values before testing your submission.

```
# Paste your code into this box
fixed = 0

x = balance
i = 1

while True:
    fixed = fixed + 10
    while i < 13:
        RemaingBalance = balance - fixed
        balance = RemaingBalance + (RemaingBalance* (annualInterestRate/12))
        i+=1
    #print balance
    if balance < 0:
        print  'Lowest Payment:', fixed
        break
```

Correct

Test results

Hide output

CORRECT

Test Case 1

balance = 3329; annualInterestRate = 0.2

Output:

```
Lowest Payment: 310
```

Test Case 2

balance = 4773; annualInterestRate = 0.2

Output:

```
Lowest Payment: 440
```

Test Case 3

balance = 3926; annualInterestRate = 0.2

Output:

```
Lowest Payment: 360
```

Randomized Test Case 1

balance = 626; annualInterestRate = 0.18

Output:

```
Lowest Payment: 60
```

Randomized Test Case 2

balance = 48; annualInterestRate = 0.25

Output:

```
Lowest Payment: 10
```

Randomized Test Case 3

balance = 539; annualInterestRate = 0.18

Output:

```
Lowest Payment: 50
```

Randomized Test Case 4

balance = 615; annualInterestRate = 0.2

Output:

```
Lowest Payment: 60
```

Randomized Test Case 5

balance = 3136; annualInterestRate = 0.04

Output:

```
Lowest Payment: 270
```

Randomized Test Case 6

balance = 3606; annualInterestRate = 0.18

Output:

```
Lowest Payment: 330
```

Randomized Test Case 7

balance = 4380; annualInterestRate = 0.15

Output:

```
Lowest Payment: 400
```

Randomized Test Case 8

balance = 3078; annualInterestRate = 0.15

Output:

```
Lowest Payment: 280
```

Randomized Test Case 9

balance = 4785; annualInterestRate = 0.18

Output:

```
Lowest Payment: 440
```

Randomized Test Case 10

balance = 3339; annualInterestRate = 0.18

Output:

```
Lowest Payment: 310
```

Randomized Test Case 11

balance = 3285; annualInterestRate = 0.04

Output:

```
Lowest Payment: 280
```

Randomized Test Case 12

balance = 3027; annualInterestRate = 0.15

Output:

```
Lowest Payment: 270
```

Hide output

Hints

Hint: How to think about this problem?

- Start with $10 payments per month and calculate whether the balance will be paid off in a year this way (be sure to take into account the interest accrued each month).
- If $10 monthly payments are insufficient to pay off the debt within a year, increase the monthly payment by $10 and repeat.

Hint: A way of structuring your code

- If you are struggling with how to structure your code, think about the following:
  - Given an initial balance, what code would compute the balance at the end of the year?
  - Now imagine that we try our initial balance with a monthly payment of $10. If there is a balance remaining at the end of the year, how could we write code that would reset the balance to the initial balance, increase the payment by $10, and try again (using the same code!) to compute the balance at the end of the year, to see if this new payment value is large enough.
  - I'm still confused!
- Be careful - you don't want to overwrite the original value of `balance`. You'll need to save that value somehow for later reference!

Reminder: Only hit "Check" once per submission. We are unable to give you more than 30 checks.

Check | Save | *You have used 2 of 30 submissions*

Your answers have been saved but not graded. Click 'Check' to grade them.

If you believe you have correct code but it is marked incorrect after clicking "Check"...

After you submit your code, you can see every test case the graders runs on your code. They compare what your code outputs with what our answer code is supposed to output. Click the small link titled "See Full Output" below the Test Results header.

Hide related resources ▲

This will be a list of resources your fellow students thought might be helpful, but it is empty currently. If you find useful resources, either on edx.org or elsewhere, please add it.

Download resources Add new resource >>

Show Discussion New Post

engineering, food and nutrition, history, humanities, law, literature, math, medicine, music, philosophy, physics, science, statistics and more. EdX is a non-profit online initiative created by founding partners Harvard and MIT.

Terms of Service and Honor Code

Privacy Policy (Revised 4/16/2014)

FAQ

edX Blog

Donate to edX

Jobs at edX

Meetup

LinkedIn

Google+