

[Courseware \(/courses/MITx/6.00x/2012\\_Fall/courseware/\)](/courses/MITx/6.00x/2012_Fall/courseware/)[Course Info \(/courses/MITx/6.00x/2012\\_Fall/info/\)](/courses/MITx/6.00x/2012_Fall/info/)[Textbook \(/courses/MITx/6.00x/2012\\_Fall/book/0/\)](/courses/MITx/6.00x/2012_Fall/book/0/)[Discussion \(/courses/MITx/6.00x/2012\\_Fall/discussion/forum/\)](/courses/MITx/6.00x/2012_Fall/discussion/forum/)[Wiki \(/courses/MITx/6.00x/2012\\_Fall/course\\_wiki/\)](/courses/MITx/6.00x/2012_Fall/course_wiki/)[Progress \(/courses/MITx/6.00x/2012\\_Fall/progress/\)](/courses/MITx/6.00x/2012_Fall/progress/)

## PROBLEM 2: PAYING DEBT OFF IN A YEAR : 15.0 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

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

```
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

**Updated balance each month** = (Previous balance - Minimum monthly payment) x (1 + Monthly interest rate)

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.

```
1 MonthlyPayment=10
2 MonthlyInterestRate= (annualInterestRate) / 12
3 p=True
4 while p==True:
5     i=1
6     PreviousBalance=balance
7     while i<=12:
8         RBalance = (PreviousBalance - MonthlyPayment) * (1 + MonthlyInterestRate)
9         PreviousBalance = RBalance
10        i=i+1
11    if PreviousBalance<=0:
12        print 'Lowest Payment: ' + str(MonthlyPayment)
13        p=False
14    MonthlyPayment=MonthlyPayment+10
```

Correct

## Test results

CORRECT

[See full output](#)

### Hints

Hint: How to think about this problem?

Hint: A way of structuring your code

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!

The autograder says, "Your submission could not be graded." Help!

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

Check

Save

*You have used 1 of 50 submissions*

Show Discussion

New Post



[Find Courses \(/courses\)](/courses) [About \(/about\)](/about) [Blog \(http://blog.edx.org/\)](http://blog.edx.org/) [Jobs \(/jobs\)](/jobs) [Contact \(/contact\)](/contact)



<http://youtube.com/user/edxonline>



<https://plus.google.com/108235383044095082735>



<http://www.facebook.com/EdxOnline>



<https://twitter.com/edXOnline>

© 2012 edX, some rights reserved.

[terms of service \(/tos\)](/tos)

[privacy policy \(/privacy\)](/privacy)

[honor code \(/honor\)](/honor)

[help \(/help\)](/help)