



[Curso](#) > [Week 2...](#) > [Proble...](#) > Proble...

Audit Access Expires Ago 5, 2020

You lose all access to this course, including your progress, on Ago 5, 2020.

Upgrade by Jul 1, 2020 to get unlimited access to the course as long as it exists on the site. [Upgrade now](#)

Problem 3

Problem 3 - Using Bisection Search to Make the Program Faster

20.0/20.0 points (graded)

You'll notice that in Problem 2, your monthly payment had to be a multiple of \$10. Why did we make it that way? You can try running your code locally so that the payment can be any dollar and cent amount (in other words, the monthly payment is a multiple of \$0.01). Does your code still work? It should, but you may notice that your code runs more slowly, especially in cases with very large balances and interest rates. (Note: when your code is running on our servers, there are limits on the amount of computing time each submission is allowed, so your observations from running this experiment on the grading system might be limited to an error message complaining about too much time taken.)

Well then, how can we calculate a more accurate fixed monthly payment than we did in Problem 2 without running into the problem of slow code? We can make this program run faster using a technique introduced in lecture - bisection search!

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

To recap the problem: we are searching for the smallest monthly payment such that we can pay off the entire balance within a year. What is a reasonable **lower bound** for this payment value? \$0 is the obvious answer, but you can do better than that. If there



was no interest, the debt can be paid off by monthly payments of one-twelfth of the original balance, so we must pay at least this much every month. One-twelfth of the original balance is a good lower bound.

What is a good **upper bound**? Imagine that instead of paying monthly, we paid off the entire balance at the end of the year. What we ultimately pay must be greater than what we would've paid in monthly installments, because the interest was compounded on the balance we didn't pay off each month. So a good upper bound for the monthly payment would be one-twelfth of the balance, *after* having its interest compounded monthly for an entire year.

In short:

Monthly interest rate = (Annual interest rate) / 12.0

Monthly payment lower bound = Balance / 12

Monthly payment upper bound = (Balance x (1 + Monthly interest rate)¹²) / 12.0

Write a program that uses these bounds and bisection search (for more info check out [the Wikipedia page on bisection search](#)) to find the smallest monthly payment *to the cent* (no more multiples of \$10) such that we can pay off the debt within a year. Try it out with large inputs, and notice how fast it is (try the same large inputs in your solution to Problem 2 to compare!). Produce the same return value as you did in Problem 2.

Note that if you do not use bisection search, your code will not run - your code only has 30 seconds to run on our servers.

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 3 Test Cases](#)

Note: The automated tests are lenient - if your answers are off by a few cents in either direction, your code is OK.

Be sure to test these on your own machine - and that you get the same output! - before running your code on this webpage!

Test Cases:



1.

Test Case 1:
 balance = 320000
 annualInterestRate = 0.2

Result Your Code Should Generate:

 Lowest Payment: 29157.09

2.

Test Case 2:
 balance = 999999
 annualInterestRate = 0.18

Result Your Code Should Generate:

 Lowest Payment: 90325.03

```

24 valor_maximo = (balance * (1 + taxa_mensal) ** 12) / 12 # B
25
26 iteracao = 1
27
28 while iteracao < 200:
29     valor_a_verificar = (valor_minimo + valor_maximo) / 2 # C
30     fc = pagando_bi(balance, annualInterestRate, valor_a_verificar)
31     if abs(fc) <= 0.01:
32         return round(valor_a_verificar, 2)
33         break
34     if sinal(fc) == sinal(pagando_bi(balance, annualInterestRate, valor_mi
35     valor_minimo = valor_a_verificar
36 else:
37     valor_maximo = valor_a_verificar
38     iteracao += 1

```

Press ESC then TAB or click outside of the code editor to exit

Correta

Test results



[Hide output](#)**CORRECT**

Test Case 1

balance = 320000; annualInterestRate = 0.2

Output:

Lowest Payment: 29157.09

Test Case 2

balance = 999999; annualInterestRate = 0.18

Output:

Lowest Payment: 90325.02

Randomized Test Case 1

balance = 65390; annualInterestRate = 0.18

Output:

Lowest Payment: 5906.36

Randomized Test Case 2

balance = 129413; annualInterestRate = 0.22

Output:

Lowest Payment: 11894.27

Randomized Test Case 3



balance = 229117; annualInterestRate = 0.22

Output:

Lowest Payment: 21058.0

Randomized Test Case 4

balance = 491597; annualInterestRate = 0.18

Output:

Lowest Payment: 44403.56

Randomized Test Case 5

balance = 424216; annualInterestRate = 0.2

Output:

Lowest Payment: 38652.83

Randomized Test Case 6

balance = 243971; annualInterestRate = 0.2

Output:

Lowest Payment: 22229.64

Randomized Test Case 7

balance = 494616; annualInterestRate = 0.2

Output:



Lowest Payment: 45067.39

Randomized Test Case 8

balance = 59243; annualInterestRate = 0.18

Output:

Lowest Payment: 5351.13

Randomized Test Case 9

balance = 442440; annualInterestRate = 0.2

Output:

Lowest Payment: 40313.32

Randomized Test Case 10

balance = 302024; annualInterestRate = 0.22

Output:

Lowest Payment: 27758.84

[Hide output](#)

Note:

Depending on where, and how frequently, you round during this function, your answers may be off a few cents in either direction. Try rounding as few times as possible in order to increase the accuracy of your result.

Important

Only hit "Check" once per submission. You only get 30 checks per problem.



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

"Staff Debug: L397 Error" means your code has an infinite loop...

Do not define your own values

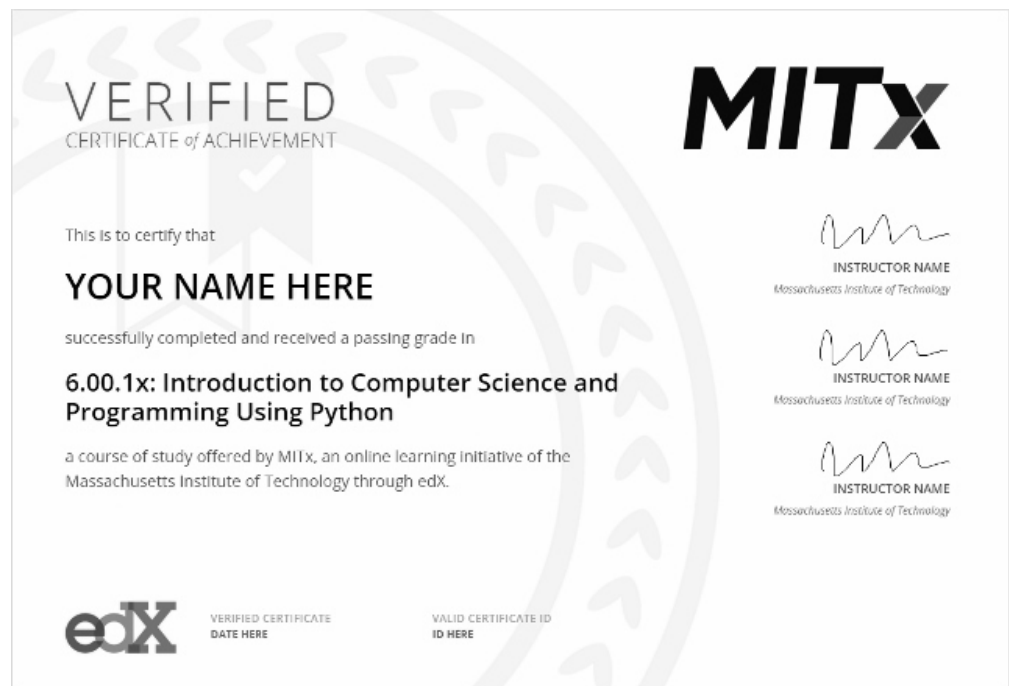
Enviar

You have used 2 of 30 attempts

Verified Certificates

Interested in using an edX certificate to bolster a college application or to advance your career? With an edX verified certificate, you can demonstrate to colleges, employers, and colleagues that you successfully completed a challenging edX


course, while helping to support the edX non-profit mission. You can register to earn a Verified Certificate directly [here](#).



Problem 3 - Using Bisection Search to Make the Program Faster

Ocultar discussão

Topic: Problem Set 2 / Problem 3

Add a l 

Show all posts ▼

por atividade recente ▼

- | | | |
|---|---|----|
| 💬 | [Spoiler] Feedback on code | 5 |
| | I have a working block of code but it seems very inefficient. For example, running it on test cas... | |
| ? | Is the interest calculated monthly on the new balance? | 2 |
| | I'm hard coding the values of the test output, and I'm not getting 0 by the end of the year. Is t... | |
| ? | how many cents off for code to be OK? | 2 |
| | how many cents off for code to be OK? | |
| 💬 | Exact solution with some algebra | 1 |
| | I understand this defeats the purpose of what this assignment is intending for us to practice,... | |
| 💬 | How long did it take you? | 25 |
| | I have been feeling discouraged about this course recently. These problems have been taking... | |
| ? | How do I write the if else statement that changes the value of balance? | 2 |
| | So far I have upperPay= max, lowerPay = min and midPay = avg between the two. My while st... | |
| 💬 | Amount of Lines taken | 4 |
| | How many lines did you guys take? I got it in 25 lines using iteration, wanted to know if there ... | |
| 💬 | Do not underestimate the power of taking a break | 4 |
| | Multiple straight hours working, 4 am, completely blocked at this point. I finally gave up for th... | |
| ? | What am I checking for? | 2 |
| | I understand my upper and lower bounds, but once I get my guess what am I checking it agai... | |
| 💬 | RecursionError | 8 |
| | I wrote a solution in PyCharm. Tested it out and for the first solution it called itself 18 times. H... | |
| 💬 | This problem is destroying my self-confidence | 9 |
| | I'm already a week behind schedule and this problem is certainly not helping me catch up. I'm... | |
| ? | EDIT: L379 NOT L397 Error for Code that Works on all Graded Cases in Spyder IDE | 5 |
| | Edited: An original iteration of my code had an error that did not result in an L379 (not 397) er... | |
| ? | Recursion | 2 |
| | Are we supposed to only use recursion on this problem? | |

© All Rights Reserved

