

For the following 3 problems, you will not be able to see the "Full Output" test cases, but you will be able to see whether you have any syntax errors. These questions will additionally test your ability to analyze what code does at a higher level.

PROBLEM 8-1 (5 points possible)

Successive approximation is a general method in which on each iteration of an algorithm, we find a closer estimate of the answer for which we are seeking. One class of successive approximation algorithms uses the idea of a fixed point. If $f(x)$ is a mathematical function, then finding the x such that $f(x) = x$ gives us the fixed point of f .

One way to find a fixed point is to start with some guess (e.g. `guess = 1.0`) and, if this is not good enough, use as a next guess the value of `f(guess)`. We can keep repeating this process until we get a guess that is within epsilon of `f(guess)`.

Here is a slightly incorrect definition of this function:

```
def fixedPoint(f, epsilon):
    """
    f: a function of one argument that returns a float
    epsilon: a small float

    returns the best guess when that guess is less than epsilon
    away from f(guess) or after 100 trials, whichever comes first.
    """
    guess = 1.0
    for i in range(100):
        if f(guess) - guess < epsilon:
            return guess
        else:
            guess = f(guess)
    return guess
```

This function definition has a bug in it. You can fix this by correcting exactly one line of the definition. Please do so in the box below.

```
1 def fixedPoint(f, epsilon):
2     """
3     f: a function of one argument that returns a float
4     epsilon: a small float
5
6     returns the best guess when that guess is less than epsilon
7     away from f(guess) or after 100 trials, whichever comes first.
8     """
9     guess = 1.0
10    for i in range(100):
11        if f(guess) - guess < epsilon:
12            return guess
13        else:
14            guess = f(guess)
15    return guess
16
```

Unanswered

Check

Save

You have used 0 of 10 submissions

PROBLEM 8-2 (5 points possible)

Assuming you have corrected the implementation of the `fixedPoint` function, we can use it to compute other useful things such as square roots. In particular, the square root of a number, `a`, is the fixed point of the function $f(x) = \frac{1}{2}(a/x + x)$.

The next two questions define two different implementations of this idea. Each has one incorrect line of code. The first is below:

```
def sqrt(a):
    def tryit(x):
        return 0.5 * (a/x + x)
    return fixedPoint(tryit(a), 0.0001)
```

This code has a bug in it. You can fix this by correcting exactly one line of the definition. Please do so in the box below.

```
1 def sqrt(a):
2     def tryit(x):
3         return 0.5 * (a/x + x)
4     return fixedPoint(tryit(a), 0.0001)
5
```

Unanswered

Check

Save

You have used 0 of 10 submissions

PROBLEM 8-3 (5 points possible)

Here is the second:

```
def babylon(a):
    def test(x):
        return 0.5 * ((a / x) + x)
    return test

def sqrt(a):
    return fixedPoint(babylon, 0.0001)
```

This code has a bug in it. You can fix this by correcting exactly one line of the definition. Please do so in the box below.

```
1 def babylon(a):
2     def test(x):
3         return 0.5 * ((a / x) + x)
4     return test
5
6 def sqrt(a):
7     return fixedPoint(babylon, 0.0001)
8
```

Unanswered

Check

Save

You have used 0 of 10 submissions



About (<https://www.edx.org/about-us>) Jobs (<https://www.edx.org/jobs>)
Press (<https://www.edx.org/press>) FAQ (<https://www.edx.org/student-faq>)
Contact (<https://www.edx.org/contact>)



EdX is a non-profit created by founding partners Harvard and MIT whose mission is to bring the best of higher education to students of all ages anywhere in the world, wherever there is Internet access. EdX's free online MOOCs are interactive and subjects include computer science, public health, and artificial intelligence.



(<http://www.meetup.com/edX-Global-Community/>)



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



(<https://twitter.com/edXOnline>)



(<https://plus.google.com/108235383044095082>)



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

© 2014 edX, some rights reserved.

[Terms of Service and Honor Code](#) -
[Privacy Policy \(https://www.edx.org/edx-privacy-policy\)](https://www.edx.org/edx-privacy-policy)