SelfAssessment92

November 2, 2015

1 Exercise 9.2:

Write a function `showIDs(seq)` that accepts a sequence `seq` as an argument. The function will print the address of the sequence and the address of every member of the sequence.

Embed your function in a program containing these lines:

```
In [ ]: import copy

        x = 2*[ 4*[1.5] ]
        y = x
        z = copy.deepcopy(x)

        print x
        showIDs(x)
        showIDs(y)
        showIDs(z)
```

Answer appears after one blank page (so you don't peek).

Are you sure you're ready to peek?

2 Possible Solution

```
In [9]: import copy

        def showIDs(x):
            print 'Object address:', id(x)
            print 'Elements:'
            for i in x: print id(i)
            print '\n'

        x = 2*[ 4*[1.5] ]
        y = x
        z = copy.deepcopy(x)

        print x,'\n'
        showIDs(x)
        showIDs(y)
        showIDs(z)
```

```
[[1.5, 1.5, 1.5, 1.5], [1.5, 1.5, 1.5, 1.5]]

Object address: 50846776
Elements:
51829736
51829736


Object address: 50846776
Elements:
51829736
51829736


Object address: 50846816
Elements:
50898304
50898304
```

In `x` the objects have the same address because the `*` method is simply creating a copy of the individual element. When this object is copied to `y` we see that it is a shallow copy, so addresses again point to the same location. `z` is a deep copy, so the address is different, but Python still recognizes that element `z[0]` and `z[1]` are the same, and attempts to minimize memory allocation by pointing to the same memory location.