

MITx: 6.00.1x Introduction to Computer Science and Programming Using Python

Courseware (/courses/MITx/6.00.1_4x/3T2014/courseware)

Updates & News (/courses/MITx/6.00.1_4x/3T2014/info)

Calendar (/courses/MITx/6.00.1_4x/3T2014/89309559b0414f6d8cbef9e48ca19f4b/) Wiki (/courses/MITx/6.00.1_4x/3T2014/course_wiki)

Discussion (/courses/MITx/6.00.1_4x/3T2014/discussion/forum) Progress (/courses/MITx/6.00.1_4x/3T2014/progress)

L11 PROBLEM 5 (5/5 points)

Consider the following code from the last lecture video:

```
class intSet(object):
    """An intSet is a set of integers
   The value is represented by a list of ints, self.vals.
   Each int in the set occurs in self.vals exactly once."""
         _init__(self):
        """Create an empty set of integers"""
       self.vals = []
   def insert(self, e):
        """Assumes e is an integer and inserts e into self"""
       if not e in self.vals:
            self.vals.append(e)
   def member(self, e):
        """Assumes e is an integer
          Returns True if e is in self, and False otherwise"""
       return e in self.vals
   def remove(self, e):
        """Assumes e is an integer and removes e from self
           Raises ValueError if e is not in self"""
            self.vals.remove(e)
       except:
            raise ValueError(str(e) + ' not found')
   def __str__(self):
        """Returns a string representation of self"""
       self.vals.sort()
       return '{' + ','.join([str(e) for e in self.vals]) + '}'
```

Your task is to define the following two methods for the <code>intSet</code> class:

1. Define an intersect method that returns a new intSet containing elements that appear in both sets. In other words,

```
s1.intersect(s2)
```

would return a new intSet of integers that appear in both s1 and s2. Think carefully - what should happen if s1 and s2 have no elements in common?

2. Add the appropriate method(s) so that <code>len(s)</code> returns the number of elements in <code>s</code>.

Hint: look through the Python docs (http://docs.python.org/release/2.7.3/reference/datamodel.html) to figure out what you'll need to solve this problem.

```
1 class intSet(object):

2 """An intSet is a set of integers

3 The value is represented by a list of ints, self.vals.

4 Each int in the set occurs in self.vals exactly once."""

5 def __init__(self):

7 """Geneta on omature of integers""
```

Correct

```
class intSet(object):
   """An intSet is a set of integers
   The value is represented by a list of ints, self.vals.
   Each int in the set occurs in self.vals exactly once."""
   def __init__(self):
        """Create an empty set of integers"""
       self.vals = []
   def insert(self, e):
       """Assumes e is an integer and inserts e into self"""
       if not e in self.vals:
           self.vals.append(e)
   def member(self, e):
       """Assumes e is an integer
          Returns True if e is in self, and False otherwise"""
       return e in self.vals
   def remove(self, e):
       """Assumes e is an integer and removes e from self
          Raises ValueError if e is not in self"""
           self.vals.remove(e)
       except:
           raise ValueError(str(e) + ' not found')
   def intersect(self, other):
       """Assumes other is an intSet
          Returns a new intSet containing elements that appear in both sets."""
       # Initialize a new intSet
       commonValueSet = intSet()
       # Go through the values in this set
       for val in self.vals:
           # Check if each value is a member of the other set
            if other.member(val):
               commonValueSet.insert(val)
       return commonValueSet
   def __str__(self):
       """Returns a string representation of self"""
       self.vals.sort()
       return '{' + ','.join([str(e) for e in self.vals]) + '}'
   def __len__(self):
        """Returns the length of the set.
          This method is called by the `len` built-in function."""
       return len(self.vals)
```

Test results

See full output

CORRECT

See full output

Check

Hide Answer





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/YourMeetup)



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



(https://twitter.com/YourPlatformTwitterAcco

(https://plus.google.com/YourGooglePlusAccc

(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)