Problem 6

Bookmark this page

Problem 6-1

0.0/15.0 points (graded)

You are given the following superclass. Do not modify this.

```
class Container(object):
    """ Holds hashable objects. Objects may occur 0 or more times """
    def __init__(self):
        """ Creates a new container with no objects in it. I.e., any object
            occurs 0 times in self. """
        self.vals = {}
    def insert(self, e):
        """ assumes e is hashable
            Increases the number times e occurs in self by 1. """
        try:
            self.vals[e] += 1
        except:
            self.vals[e] = 1
    def __str__(self):
        s = ""
        for i in sorted(self.vals.keys()):
            if self.vals[i] != 0:
                s += str(i)+":"+str(self.vals[i])+"\n"
        return s
```

Write a class that implements the specifications below. Do not override any methods of Container.

```
class Bag(Container):
    def remove(self, e):
        """ assumes e is hashable
            If e occurs in self, reduces the number of
            times it occurs in self by 1. Otherwise does nothing. """
        # write code here

    def count(self, e):
        """ assumes e is hashable
            Returns the number of times e occurs in self. """
        # write code here
```

- For example,

  ```
  d1 = Bag()
  d1.insert(4)
  d1.insert(4)
  print(d1)
  d1.remove(2)
  print(d1)
  ```

  prints

  ```
  4:2
  4:2
  ```

- For example,

  ```
  d1 = Bag()
  d1.insert(4)
  d1.insert(4)
  d1.insert(4)
  print(d1.count(2))
  print(d1.count(4))
  ```

  prints

  ```
  0
  3
  ```

Paste your entire class, including the definition, in the box below. Do not leave any debugging print statements.

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

Unanswered

Submit You have used 0 of 10 attempts

Save Reset

Problem 6-2

0.0/5.0 points (graded)

Write a method in `Bag` such that if `b1` and `b2` were bags then `b1+b2` gives a new bag representing the union of the two bags.

- For example,

  ```
  a = Bag()
  a.insert(4)
  a.insert(3)
  b = Bag()
  b.insert(4)
  print(a+b)
  ```

  prints

  ```
  3:1
  4:2
  ```

Paste your entire class for `Bag` with the new method, including the definition, in the box below. Do not leave any debugging print statements.

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

Unanswered

Submit You have used 0 of 10 attempts

Save Reset

Problem 6-3

0.0/15.0 points (graded)

Write a class that implements the specifications below. Do not override any methods of Container.

```
class ASet(Container):
    def remove(self, e):
        """assumes e is hashable
           removes e from self"""
        # write code here

    def is_in(self, e):
        """assumes e is hashable
           returns True if e has been inserted in self and
           not subsequently removed, and False otherwise."""
        # write code here
```

- For example,

```
d1 = ASet()
d1.insert(4)
d1.insert(4)

d1.remove(2)
print(d1)

d1.remove(4)
print(d1)
```

prints

```
4:2 # from d1.remove(2) print
```

```
    # (empty) from d1.remove(4) print
```

- For example,

  ```
  d1 = ASet()
  d1.insert(4)
  print(d1.is_in(4))
  d1.insert(5)
  print(d1.is_in(5))
  d1.remove(5)
  print(d1.is_in(5))
  ```

  prints

  ```
  True
  True
  False
  ```

Paste your entire class, including the definition, in the box below. Do not leave any debugging print statements.

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