******Counter******

Counter类的目的是用来跟踪值出现的次数。它是一个无序的容器类型，以字典的键值对形式存储，
其中元素作为key，其计数作为value。计数值可以是任意的Interger（包括0和负数）。
Counter类和其他语言的bags或multisets很相似。

```
c = Counter('abcdeabcdabcaba')
print c
```
输出：Counter({'a': 5, 'b': 4, 'c': 3, 'd': 2, 'e': 1})

****创建

Counter类创建的四种方法：

```
>>> c = Counter()  # 创建一个空的Counter类
>>> c = Counter('gallahad')  # 从一个可iterable对象（list、tuple、dict、字符串等）创建
>>> c = Counter({'a': 4, 'b': 2})  # 从一个字典对象创建
>>> c = Counter(a=4, b=2)  # 从一组键值对创建
```

****计数值的访问与缺失的键

当所访问的键不存在时，返回0，而不是KeyError；否则返回它的计数。

计数值的访问
```
>>> c = Counter("abcdefgab")
>>> c["a"]
>>> c["c"]
>>> c["h"]
```

****计数器的更新（update和subtract）

可以使用一个iterable对象或者另一个Counter对象来更新键值。
计数器的更新包括增加和减少两种。其中，

增加使用update()方法：
计数器的更新（update）
```
>>> c = Counter('which')
>>> c.update('witch')  # 使用另一个iterable对象更新
>>> c['h']
>>> d = Counter('watch')
>>> c.update(d)  # 使用另一个Counter对象更新
>>> c['h']
```

减少则使用subtract()方法
计数器的更新（subtract）
```
>>> c = Counter('which')
>>> c.subtract('witch')  # 使用另一个iterable对象更新
>>> c['h']
>>> d = Counter('watch')
>>> c.subtract(d)  # 使用另一个Counter对象更新
>>> c['a']
-1
```

*****键的修改和删除

当计数值为0时，并不意味着元素被删除，删除元素应当使用del。
键的删除

```
>>> c = Counter("abcdcba")
>>> c
Counter({'a': 2, 'c': 2, 'b': 2, 'd': 1})
>>> c["b"] = 0
>>> c
Counter({'a': 2, 'c': 2, 'd': 1, 'b': 0})
>>> del c["a"]
>>> c
Counter({'c': 2, 'b': 2, 'd': 1})
```

****elements()

返回一个迭代器。元素被重复了多少次，在该迭代器中就包含多少个该元素。
元素排列无确定顺序，个数小于1的元素不被包含。

elements()方法
```
>>> c = Counter(a=4, b=2, c=0, d=-2)
>>> list(c.elements())
['a', 'a', 'a', 'a', 'b', 'b']
```

****most_common([n])

返回一个TopN列表。如果n没有被指定，则返回所有元素。
当多个元素计数值相同时，排列是无确定顺序的。

most_common()方法
```
>>> c = Counter('abracadabra')
>>> c.most_common()
[('a', 5), ('r', 2), ('b', 2), ('c', 1), ('d', 1)]
>>> c.most_common(3)
[('a', 5), ('r', 2), ('b', 2)]
```

****浅拷贝copy

浅拷贝copy
```
>>> c = Counter("abcdcba")
>>> c
Counter({'a': 2, 'c': 2, 'b': 2, 'd': 1})
>>> d = c.copy()
>>> d
Counter({'a': 2, 'c': 2, 'b': 2, 'd': 1})
```

****算术和集合操作

+、-、&、|操作也可以用于Counter。其中&和|操作分别返回两个Counter对象各元素的最小值和最大值。
需要注意的是，得到的Counter对象将删除小于1的元素。

Counter对象的算术和集合操作
```
>>> c = Counter(a=3, b=1)
>>> d = Counter(a=1, b=2)
>>> c + d  # c[x] + d[x]
Counter({'a': 4, 'b': 3})
>>> c - d  # subtract（只保留正数计数的元素）
Counter({'a': 2})
>>> c & d  # 交集:  min(c[x], d[x])
Counter({'a': 1, 'b': 1})
>>> c | d  # 并集:  max(c[x], d[x])
Counter({'a': 3, 'b': 2})
```

****其他常用操作

下面是一些Counter类的常用操作，来源于Python官方文档

Counter类常用操作
```
sum(c.values())  # 所有计数的总数
c.clear()  # 重置Counter对象，注意不是删除
list(c)  # 将c中的键转为列表
set(c)  # 将c中的键转为set
dict(c)  # 将c中的键值对转为字典
c.items()  # 转为(elem, cnt)格式的列表
Counter(dict(list_of_pairs))  # 从(elem, cnt)格式的列表转换为Counter类对象
c.most_common()[:-n:-1]  # 取出计数最少的n个元素
c += Counter()  # 移除0和负值
```