


```
class Bag:
```

```
    def __init__(self):  
        self.__elems = []
```

```
    def add(self, e):  
        self.__elems.append(e)
```

```
    def remove(self, e):  
        if e in self.__elems:  
            self.__elems.remove(e)  
            return True  
        return False
```

```
    def nrOccurrences(self, e):  
        count = 0  
        for i in self.__elems:  
            if i == e:  
                count += 1  
        return count
```

```
def size(self):  
    return len(self.__elems)
```

```
def search(self, e):  
    return e in self.__elem
```

```
def iterator(self):  
    return BagIterator(self)
```

```
class BagIterator:
```

```
def __init__(self, b)  
    self.__bag = b  
    self.__current = 0
```

```
def next(self):  
    if self.__current < self.__bag.size():  
        self.__current += 1
```

```
    else:  
        raise ValueError()
```

```
def getcurrent(self):  
    if self.__current >= self.__bag.size():  
        raise ValueError()  
    return self.__bag.__elems[self.__current]  
  
def valid(self):  
    return self.__current < self.__bag.size()-1  
  
def first(self):  
    self.__current = 0
```

```
class BagF:
```

```
    def __init__(self):  
        self.__elems = []  
        self.__freq = []  
  
    def add(self, el):  
        if el in self.__elems:  
            index = self.__elems.index(el)  
            self.__freq[index] += 1  
        else:  
            self.__elems.append(el)  
            self.__freq.append(1)
```

```
def remove (self, e):  
    if e is not in self.__elems:  
        return False  
    else:  
        index = self.__elems.index(e)  
        if self.__freq[e] == 1  
            self.__elems.pop(index)  
            self.__freq.pop(index)  
        else:  
            self.__freq[index] -= 1  
    return True
```

```
def size (self):  
    S = 0  
    for x in self.__freq:  
        S += x  
    return S
```

class BagIterator:

def __init__(self, b):

self.__bag = b

self.__currentE = 0

self.__currentF = 1

def valid(self):

if self.__currentE >= len(self.__bag.BagF.__elems):

return False

return True

def __next__(self):

if len(self.__bag.__elems) < self.__currentE:

raise ValueError()

else:

if self.__currentF == self.__bag.__bagE[

self.__currentE]:

self.__currentE += 1

self.__currentF += 1

else:

self.__currentF += 1

```
def getCurrent(self):  
    if self.valid():  
        return self._bag._elems[self._current]  
    else:  
        raise ValueError
```