CSC 242

Lab 5

Option 1:

The **remove** operation performs two searches of a bag: one during the test of the method's precondition (using the **in** operator) and the other to locate the position of the target item to actually remove it. One way to eliminate the redundant search is to track the position of the target item in an instance variable. In the case of an array-based bag, this position would be **-1** at startup and whenever a target item is not found. If the **in** operator finds a target item, the position variable is set to that item's index in the array. After the **remove** method checks its precondition, no search loop is necessary; the method can just close the hole in the array using the position variable. Modify the **ArrayBag** class to support this capability. Note that you will now have to add a **__contains__** method to **ArrayBag** that performs this customized search.

Tip:

Your `remove` function has this code already:

```
if not item in self:
    raise KeyError(str(item) + " not in bag")
```

The use of "`in`" above will call the arraybag `__contains__` function where the instance variable, let's call it `self._targetIndex`, will get initialized to the actual index of the item if it is found and will return `True`. Otherwise `self._targetIndex` will stay at `-1` and the `KeryError` will be raised because the `_contains__` function will return `False`.
After the `if` statement above, `self._targetIndex` will be initialized to the appropriate index, so you won't have to search for the desired item again and can immediately start shifting the items over to cover up the removed index.

Write a driver program to test this newly implemented **remove** method.

Option 2:

Add the **^** operator to the language of expressions processed by the expression evaluator of the **Chapter 7 Case Study**. This operator has the same semantics as Python's exponentiation operator ******. For an example, the expression: **2 4 3 * ^** evaluates to **4096**.

Write a driver program to test this newly implemented operator on any given expression.