Test 2 Booster: Compsci 101

Owen Astrachan and Kristin Stephens-Martinez

April 24, 2018

Name: ____________________________________________

NetID/Login: ____________

Section Number: (01-Astrachan, 02-Stephens-Martinez): ____________

Honor code acknowledgment (signature) _________________________________

<table>
<thead>
<tr>
<th></th>
<th>value</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1</td>
<td>6 pts.</td>
<td></td>
</tr>
<tr>
<td>Problem 2</td>
<td>4 pts.</td>
<td></td>
</tr>
<tr>
<td>TOTAL:</td>
<td>10 pts.</td>
<td></td>
</tr>
</tbody>
</table>

In writing code you do not need to worry about specifying the proper import statements. Don’t worry about getting function or method names exactly right. Assume that all libraries and packages we’ve discussed are imported in any code you write.

Be sure your name and net-id are legible on this page and that your net-id appears at the top of every page.
PROBLEM 1:  *(HPV, Tdap, MMR)*

You’re given a dictionary in which keys are country names and values are the list of colors of that country’s flag. For example a partial list of a dictionary for flags/colors is shown below.

```python
flags = {
    "Dominica" : ["gold", "white", "black", "red", "green"],
    "St. Lucia" : ["gold", "white", "black", "blue"],
    "Estonia" : ["white", "black", "blue"],
    "Australia" : ["white", "red", "blue"],
    "Iceland" : ["white", "red", "blue"],
    "Ghana" : ["gold", "black", "red", "green"],
    "Uruguay" : ["gold", "white", "blue"],
    "Ethiopia" : ["gold", "red", "blue", "green"]
}
```

Note that `flags["Ghana"]` is a list of four colors ["gold", "black", "red", "green"], and that `flags["Iceland"][1]` is the string red.

**Part A (4 points)**

Write the function `colors` that takes a dictionary in the format shown above as a parameter and returns a new dictionary in which keys are color names (strings) and the corresponding value is a list of countries that have that color in the flag. For the dictionary shown above the call `colors(flags)` returns this dictionary

```python
{'gold' : ['Dominica', 'St. Lucia', 'Ghana', 'Uruguay', 'Ethiopia'],
 'white' : ['Dominica', 'St. Lucia', 'Estonia', 'Australia', 'Iceland', 'Uruguay'],
 'black' : ['Dominica', 'St. Lucia', 'Estonia', 'Ghana'],
 'red' : ['Dominica', 'Australia', 'Iceland', 'Ghana', 'Ethiopia'],
 'green' : ['Dominica', 'Ghana', 'Ethiopia'],
 'blue' : ['St. Lucia', 'Estonia', 'Australia', 'Iceland', 'Uruguay', 'Ethiopia']
}
```

Complete the function on the next page
def colors(flags):
    """
    returns a reverse dictionary based
    on keys and values in flags, a dictionary
    """

    newd = {}

    return newd

(see next page for another problem)
Part B

Write the function `topcolors` whose parameter is a dictionary in the format returned by `colors`, e.g.,

```python
{'gold' : ['Dominica', 'St. Lucia', 'Ghana', 'Uruguay', 'Ethiopia'],
 'white' : ['Dominica', 'St. Lucia', 'Estonia', 'Australia', 'Iceland', 'Uruguay'],
 'black' : ['Dominica', 'St. Lucia', 'Estonia', 'Ghana'],
 'red' : ['Dominica', 'Australia', 'Iceland', 'Ghana', 'Ethiopia'],
 'green' : ['Dominica', 'Ghana', 'Ethiopia'],
 'blue' : ['St. Lucia', 'Estonia', 'Australia', 'Iceland', 'Uruguay', 'Ethiopia']
}
```

The function `topcolors` returns a list of tuples of the form `(string, int)` where the string is the name of a color (a key in the dictionary parameter) and the int is the size of the corresponding value, the number of countries whose flag contains that color. The tuples are sorted from highest to lowest based on the number of countries whose flag contains the color. Ties are broken alphabetically.

For the dictionary shown above the list of tuples returned should be as shown below. Note that `('blue', 6)` appears before `('white', 6)` because `blue` is before `white` alphabetically.

```python
[('blue', 6), ('white', 6), ('gold', 5), ('red', 5), ('black', 4), ('green', 3)]
```

Complete the function below

```python
def topcolors(d):
    """
    d is a dictionary with keys that are strings
    and values are list of strings
    return a list of tuples (key,count) where
    key is a key in parameter d and count
    the length of the corresponding value d[key]
    sort by size of list, break ties alphabetically by key
    """
```