

CS1110 Lab 7 (Mar 24-25, 2015)

First Name: _____ Last Name: _____ NetID: _____

The lab assignments are very important and you must have a CS 1110 course consultant “tell CMS” that you did the work. (Correctness does not matter.) This can be done any time up until the start of the next lab (Apr 7-8). Thus, if you have trouble with a problem, then you have 1 week to get help from the teaching staff. If you finish before the hour is over, then you can leave early or you can work on the current assignment. Indeed, you are not required to physically attend the labs at all. Just make sure your work is “checked off” by a consultant. And remember this: *The lab problems feed into the assignments and the assignments define what the exams are all about.*

1 Getting Set Up

Review Lectures 14 (Lists), 15 (Lists and Functions), and 16 (More Lists and Function). From the Lab webpage download `ShowNestedLoops.py`, `ShowSelect.py`, and `ShowCities.py`. Put them all in the same folder, say `Lab7`. In the command shell, navigate the file system so that this folder is THE CURRENT WORKING DIRECTORY.

2 Nested Loops and Lists of Strings

Consider the module `ShowNestedLoops.py`:

```
def AminoAcids(alpha):
    """
    PreC: alpha is a nonempty string.
    """
    n = len(alpha)
    for i in range(n):
        for j in range(n):
            for k in range(n):
                w = alpha[i]+alpha[j]+alpha[k]
                print w

if __name__ == '__main__':
    AminoAcids('AGCT')
```

(a) Without running the application script, describe what the function `AminoAcids` does.

(b) Modify `AminoAcids` so that the output strings are numbered, e.g.,

```
0   AAA
1   AAG
2   AAC
etc
```

After you make that change, what is the last line of output?

(c) Remove the `print` statement from `AminoAcids` and then add code so that it returns a list that consists of all 3-character strings made of up characters from `alpha`. After you make that change to `AminoAcids`, how would you complete the following application script so that when it is run, exactly the same output is produced as in part (b):

```
if __name__ == '__main__':  
    a = AminoAcids('AGCT')
```

(d) We say that a length-3 string is a “no-repeat string” if each of its characters appears exactly once. Complete the following application script so that it creates a second list `b` that consists of all no-repeat strings whose characters come from `'AGCT'`

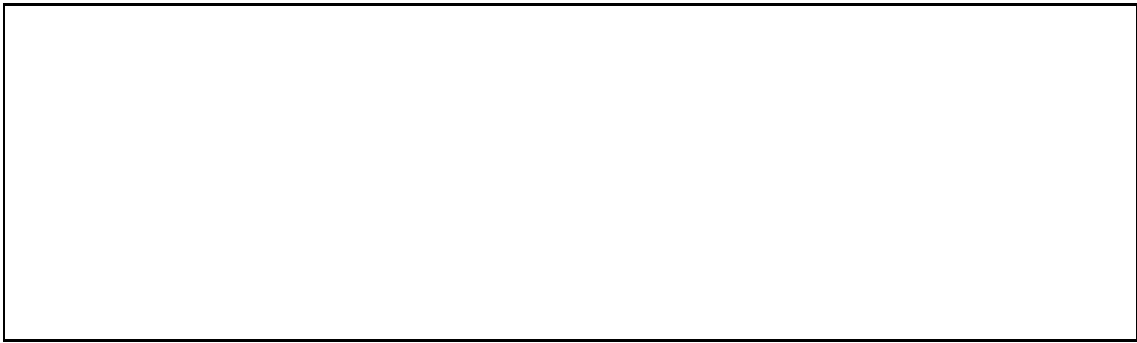
```
if __name__ == '__main__':  
    a = AminoAcids('AGCT')
```

3 Subscribing Practice

Consider the module `ShowCities.py`:

```
def MLB():
    """ Returns (City,x,y) where Cities is a list of strings that names
        28 MLB cities and lists of floats x and y that specify their xy location.
    """
if __name__ == '__main__':
    (C,x,y) = MLB()
```

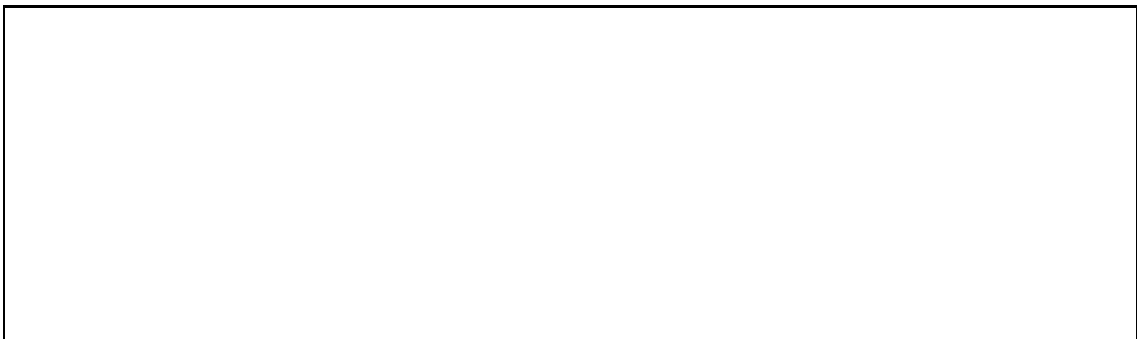
(a) Complete the application script so that it prints a list of the cities in reverse alphabetical order. To make it interesting, do not use `sort`. Note that `MLB` returns an alphabetized list of the cities.



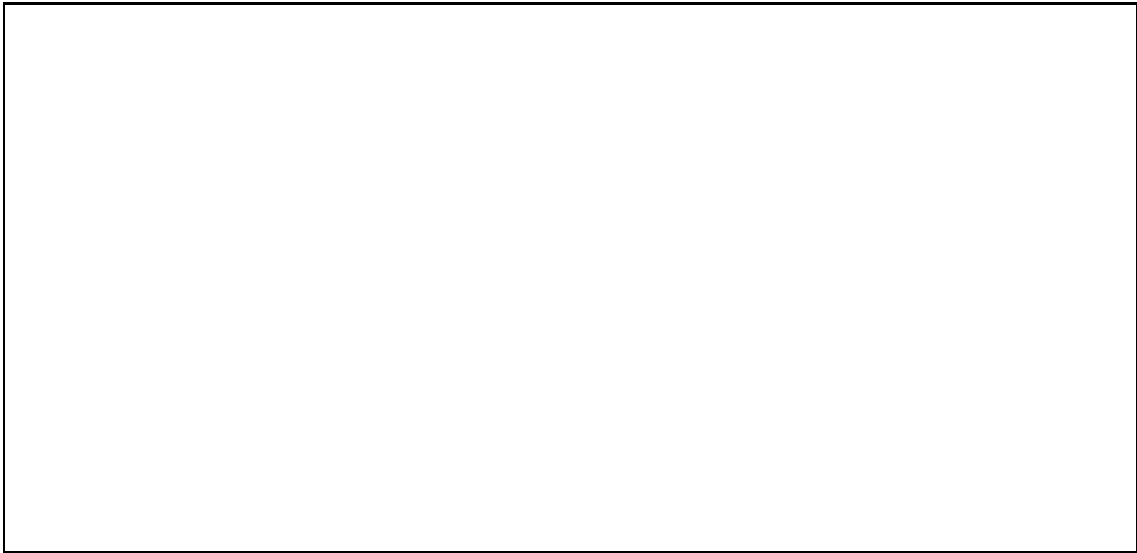
(b) Complete the application script so that it prints a list of all those cities that are within 400 miles of (300,200).



(c) Complete the application script so that it prints average length of a city string.



(d) Complete the application script so that it prints average the distance between two different cities.



Make sure you don't double count, i.e., your running sum should not include the distance from Boston to Baltimore AND the distance from Baltimore to Boston.

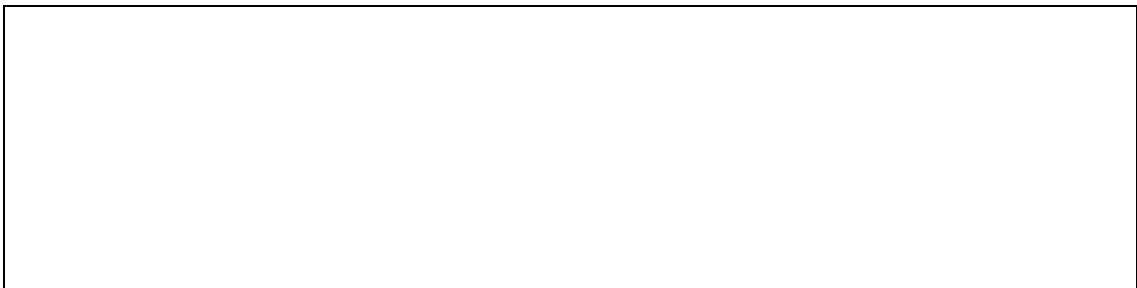
4 Index Lists

Assume that `i` and `s` are initialized as follows:

```
i = [3,2,4,0]
s = ['a','b','c','d']
```

Answer these questions without using computer. Then confirm your answer with the computer.

(a) What is the value of `s[i[0]]+s[i[1]]+s[i[2]]+s[i[3]]`?



(b) What is the value of the expression `i[i[i[2]]]`?



5 Selection Sort

Run `ShowSelect` and study the function `Select(x,i)`.

(a) Give an example of a length-5 list `a` so that if these two statements are executed

```
b = list(a)
Select(a,2)
```

then `b==a` is `True`.

(b) Rewrite the function `SelectionSort(a)` so that it returns a sorted version of `a` without changing `a`, i.e., implement this

```
def SelectionSort(a):
    """ Returns a list that is obtained by sorting the elements of
        a from smallest largest. The list a is not changed.

        PreC: a is a nonempty list of numbers.
    """
```

Give an example of a length-5 list `a` where `list(a) == Select(a,2)` is `True`.