

CSE/ISE 337 Assignment 1

Due date: Tuesday, Feb. 23, at 11:55pm

Important! Must read: (a) When writing programs, you **must** use the techniques that are described in the lecture notes and illustrated in the given samples. You may **not** use methods, modules, packages that were not covered in lectures. (b) You **must** first read the lecture slides “0-Course-Overview.pdf” available in Blackboard – Documents – Lecture Slides, especially those related to assignments, namely, Slides 0-9 to 0-14, and follow them. (c) Start working on this assignment right away; you will **not** be able to finish it if you wait until the last day.

1. (8pts) Write a Python program that removes all pairs of “” and “” from an input file. You may assume that they are always in pairs. The name of the input file is obtained from user prompt; the file name always has “.html” as suffix. After removal, the output should be written to a file whose name is obtained by appending the string “_output” to the input file name. For example, a sample input file “announce.html” reads as follows:

```
<p>Use the Web page to find the schedule for this course.
<span><span>http</span></span>://<span><span>www</span></span>3.cs.<span><s
pan>stonybrook</span></span>.<span><span>edu</span></span>/<span><span>~ell
enyliu</span></span>/337/. Python documentation can be found at:
<span><span>https</span></span>://docs.python.<span><span>org</span></span></span>
/2/.</p>
```

The corresponding output file should be called “announce_output.html”, and should read as:

```
<p>Use the Web page to find the schedule for this course.
http://www3.cs.stonybrook.edu/~ellenyliu/337/. Python documentation can be
found at: https://docs.python.org/2/.</p>
```

2. (8pts) Write a python program to decode the following encrypted English text, given the following **hints about the encryption scheme** used: (a) letter ‘z’ is encrypted to letter ‘v’, letter ‘y’ is encrypted to letter ‘u’, letter ‘b’ is encrypted to letter ‘x’, and letter ‘a’ is encrypted to letter ‘w’. (b) All non-alphabetic characters are not encrypted. You may hard code the encrypted text in your code. Output should be printed to stdout. Solution hints: you should first use the given hints to figure out the encryption scheme used, then, derive the corresponding decryption scheme. Consider using the chr() and ord() functions.

```
pda lupdkj lnkcniwiejc hwjcwca swo ejrajpaaz xu cpezk rwj nkooqi, w zqpyd
ykilqpan lnkcniwian, wxkqp 25 uawno wck. rwj nkooqi zabejaz deo ckwho bkn
lupdkj wo bkhhkso: wj awou wjz ejpqpopera hwjcwca fqop wo lksanbqh wo iwfk
ykilapepkno; klaj okqnya, ok wjukja ywj ykjpnextpa pk epo zarahklijp; ykza
pdwp eo wo qjzanopwjzwxha wo lhwej ajcheod; oqepwxehepu bkn aranuzwu pwogo,
whhksejc bkn odknp zarahklijp peiao
```

3. (6pts) In Python II slide on “Assignment and references”, to make a true copy of a list, we used a slice. Examine the following. i. What do you predict will be printed as the value of c?

```
>>> a=[1,2]
>>> b=[a,3]
>>> c=b[:]
>>> a[0]=7
>>> b[1]=8
>>> c
```

- ii. Try executing these statements. Did the result match your prediction? Explain the outcome using the concept of references.
- iii. To make a true and complete copy of a multi-level value, you can use the module function copy.deepcopy(x). Show its outcome in the example given.
- iv. Come up with an example to make a deep copy of a **three-level** value. Show its outcome.

4. (9pts) The function `randrange()` from the `random` module can be used to produce an integer from a given range. For example, `random.randrange(1, 5)` produces the values 1 to 4 inclusive with equal probability. Given the following python program, i. determine the next few lines to be executed after line #7 if `n` and `r` are of the following values as the result of execution. Please number your answers clearly. (a) if `n = 2` and `r = 1`. (b) if `n = 1` and `r = 1`. (c) if `n = 2` and `r = 2`. (d) if `n=0` and `r=2`. ii. In general, when will the statement “Wow, you are lucky” be printed?

```
1  #!/usr/local/bin/python
2  from random import randrange
3
4  for n in range(3):
5      print n
6      r = randrange(0,3)
7      print r
8      if n==r: continue
9      if n>r: break
10     print "x"
11 else:
12     print "Wow, you are lucky\n"
13
14 if n<2:
15     print "Better luck next time\n"
```

5. (5pts) Comparison of two strings is based on the ASCII ordering. This means that uppercase letters (all of them) come before lowercase letters. Many applications require a sort that is case-insensitive. Show how to achieve this effect by writing a comparison function for the sort operation (see `sortList.py` in Python II section of the sample code page for an example of such a comparison function). Ask the program user to enter four strings from stdin, one string per line. Sort these strings case-insensitively and print them to the stdout in the sorted order, with one string per line. Hints: Consider using the `lower()` function.
6. (10pts) Given two input files that contain customer records. Use Python dictionary to find (a) those customers who are only in the first (old) file, (b) those who are only in the second (new) file, and (c) those who are in both, respectively. Each input line corresponds to one customer record, which contains five tab-separated fields: Last name, First name, Email address, Customer number, and Phone number, all in double quotes, in that order. The first, second, and fourth fields are always non-empty. The other two fields may contain null values, denoted by `""`. Below are two sample input records:

"Zhuang"	"Joseph"	"jzhuang@hotmail.com"	"71303869"	"616-203-4435"
"Johnson"	"Michel"	"mjohnson@yahoo.com"	"54810672"	"204-597-1505"

The names of the two input files are two command line arguments. The output for (a), (b), and (c) are in three files named “old”, “**new**”, and “**current**” respectively. The output should contain all five fields. For (c), the new record data in the second input file should be used. Note that multiple customers may have identical last names. We assume that when both last and first names are taken into account, all customer names are unique.

7. (11pts) We would like to process some real Web content. Go to <http://adm-kng.com/archive>. You will see many images. When hovering the mouse pointer over an image, you may see a date and a note count. E.g., “Jan 22, 2016. 102 notes”. We would like to find out the five highest note counts on this page. First, learn how to “view page source” in your browser, and copy and paste the page source into a file. A sample source file is provided to you in the assignment folder, called “Landscapes_viewsouce.html”. You should study the source

format, then write a Python script to extract all note counts and find out the five highest counts. Your script should first prompt the user for the name of a page source file. It should print five numbers, corresponding to the top five note counts.

(Bonus 6pts) Each image is also clickable. The URL when an image is clicked is also in the page source. As a bonus, produce a simple HTML page that contains the URLs for the five images that have the highest notes counts, in the following format: (only 3 URLs are shown)

```
<html><body>
#1: <a href=http://adm-kng.com/post/136730014976/edmonton-sunset-by-adam-king-instagram>2,132
notes</a><br><br>
#2: <a href=http://adm-kng.com/post/138875309721/adm-kng-lake-louise-by-adam-king-
instagram>1,596 notes</a><br><br>
#3: <a href=http://adm-kng.com/post/136630625356/canmore-by-adam-king-instagram>989
notes</a></body></html>
```

8. (5pts) List comprehension. i. Define a function that takes two lists as input. It returns the difference of two lists as a list comprehension. The difference consists of the elements in the first list that are not part of the second. ii. Using the difference function you wrote in the i, define a function that returns the union of two sets that are represented by lists. A union contains all elements that are found in either list, but if the same element is found in both lists, it is included only once. iii. Illustrate the correctness of your functions by examples.

Deliverables

(3pts for right submission format) Your assignment submission should include three files: (a) a printout of all Python programs that you write for the coding questions/parts. Concatenate them into one **plaintext** file called “**a1-printout.txt**”. Each program should be clearly labeled with its corresponding question/part numbers. (b) Your written answers together with the readme information for your programs. Be sure to indicate the following information at the start of this file: *course number, assignment number, your name, and ID number, python version used, platform used*. This file should briefly describe how to use your submitted programs, and certain test cases that you recommend. It must be in **pdf**. Name this file “**a1-written.pdf**”. (c) A **zip** file that includes all individual programs that you write. Name it “**a1-source.zip**”. Be sure to name each program using its question and part number, e.g., “q1.py”, “q7withBonus.py”, and so on. You should include certain amount of program documentation, i.e., in-line comments, in your programs for important steps used. Do not repeat what the line of code says; rather write comments to help readers to understand your code.

Total: 65 points

Submission instructions

The handing-in will be through Blackboard Assignment. The submission instructions are at: <http://it.stonybrook.edu/help/kb/creating-and-managing-assignments-in-blackboard>.

You **must** read the submission instructions very carefully, and check to make sure your assignment has been submitted correctly **before** the deadline.

You can only submit once! However you can save your work by clicking "Save" as many times as you like. Only click "Submit" after you have checked and are certain that all requirements are followed.

Late submissions will not be accepted. The due date is **11:55pm on Tuesday, Feb. 23**.