

CSE/ISE 337 Assignment 1 (Spring 2014)

Due date: Thursday, Feb. 27, 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 on Slides 0-9 to 0-13, 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. (10pts) Write a Python program that will read a set of numbers from the program user with one number per line, and then print the second largest number in the set. Use -99 to signify the end of user input. Your program should be able to handle both integers and floating-point numbers. -99 is not in the set. Hint: use the input() function to read user input.
2. (10pts) Write a python program to decode the following encrypted English text, given the following hints about the encryption scheme used: (a) letter 'k' is encrypted to letter 'n', letter 'r' is encrypted to letter 'u', and letter 'x' is encrypted to letter 'a'. (b) Punctuation marks are not encrypted. You can hard code the encrypted text in your code. Hint: use the ord() and chr() functions that were described in class.

```
fjxdfkb qebob'p kl zlrkqofbp fq fpk'q exoa ql al klqefkd ql hfii lo afb clo
xka kl obifdflk qll fjxdfkb xii qeb mblmib ifsfkd ifcb fk mbxzb vlr jxv pxv
qexq F'j x aobxjbo yrq F'j klq qeb lkiv lkb f elmb pljbaxv vlr'ii glfk rp
xka qeb tloia tfii yb xp lkb
```

3. (9pts) Given the following python program, 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 = 0 and r = 0. (b) if n = 1 and r = 2. (c) if n = 2 and r = 2

```
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"
```

4. (9pts) (a) Explain the difference between the second and third expressions in the left column below:

```
>>> lst = [1, 2, 3]
>>> lst * 3
...
>>> [ lst ] * 3
```

```
>>> lst = [1, 2, 3]
>>> arr = [ lst ] * 3
>>> lst[1] = 7
>>> print arr # what do u see?
```

```
>>> lst = [1, 2, 3]
>>> arr = [ lst[:] ] * 3
>>> arr[1][1] = 7
>>> print arr # what do u see?
```

(b) Take the third expression in (a), and subsequently change the value `lst`; see the middle column above, what happens to the result? Explain the behavior.

(c) What if you tried to avoid the problem above using the slice copy, see the right side column above? What happens if you change an element in the array? Explain the result.

5. (10pts) Write a program that will take a string as input. The input contains all 3-digit course numbers. Your program removes all repetitive course numbers from the input, classifies the courses according to the first digits in the course numbers, and outputs the number of courses per category. For example, if the input is `230 104 316 110 104 126`, the output should be `100 : 4 200 : 1 300 : 1`. It says the input has four unique 100-level courses, one 200-level course, and one 300-level course.
6. (Bonus 8pts) The directory listing command “`ls -l`” on Linux/UNIX/Mac OS displays the content of the current directory where this command is run. Write a program that takes a file as input. This file contains the “`ls -l`” command output of some directory. Your program prompts user for a month and a year, and finds all those files in that directory that are dated with the given month and year whose file sizes are larger than 12000 bytes. For example, the following directory content contains six files. The first file `test1` is dated Nov 17, 2011, with file size 16858 bytes. The second file `PaperListing2011` is dated Nov 17, 2011, with file size 1344. You may assume that all lines are such description of files, no sub-directories.

```
-rw-r--r--  1 jsmith  student      16858 17 Nov  2011 test1
-rw-r--r--  1 jsmith  student       1344 17 Nov  2011 PaperListing2011
-rw-r--r--  1 jsmith  student      35840  9 Jul  2012 2011.doc
-rwxrwxrwx  1 jsmith  student    2763443  4 Apr  2012 IMG_9090.JPG
-rw-r--r--  1 jsmith  student      1484 24 Mar  2013 goldfish.txt
-rw-r--r--  1 jsmith  student       163 17 Jan  2013 testtk.py
```

If a user enters `Nov` and `2011` as given month and year, the output of your program should be:

```
Which month are you interested in? Nov
Which year are you interested in? 2011
-rw-r--r--  1 jsmith  student      16858 17 Nov  2011 test
```

7. (10pts) Functional programming. (1) Create a list of values from 2 to 1000. Use the filter function to eliminate all list elements that are multiples of the first element, and print the remaining list elements. Hint: you may want to define a lambda function that determines if an input number is a multiple of the first list element
- (2) Continue from Part (1), write a recursive function that finds prime numbers using the following method: start with the list of the numbers from 2 to 1000, written in order. Select the first element, which will be a prime number. Then remove all the values that are a multiple of the first element. Repeat until the list is empty. This function takes a list as input and prints all prime numbers within the list. A comma followed by a space should separate the output numbers. Hint: use the filter function that you wrote in Part (1)

Deliverables

Your assignment submission should include three files: (a) a printout of all programs that you write in questions 1,2,5,6,7. Concatenate them into one **plaintext** file called “**a1-printout.txt**”. Each program should be clearly labeled with its corresponding question/part numbers. (b) Your

answers to questions 3 and 4, 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 release 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., “q1part1”, “q1part2”, 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. (2pts for right submission format)

Total: 60 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 Thursday, Feb. 27**.