CPSC 231 – Fall 2012, L01/L03
Assignment 3: Histogram for Grade Distribution

Due: *Friday November 09 2012, 4:00pm*

In this assignment, you will exercise your Python skills on string and list manipulation. You will also need previous tools such as iteration and functions.

Histogram plotting for grade distribution

Your task is to write a Python program that processes a text file containing grades for a class, extracts the desired grades, count the number of grades in each grade segment, and then visualizes the distribution with the help of QuickDraw.

Reading the input

The input is a file of the following format:

```
ID      , Last  ,  First, Lecture, Tutorial, A1, A2, A3, A4, A5
10034567, Smith,   Winston,    L01, T03, 6, 5.5, 8, 10, 8.5
10045678, Lee,     Bruce,      L02, T05, 4.5, 6.5, 7, 7, 8.5
00305678, Obama,   Jack,       L01, T05, 10, 10, 9, 9.5, 10
00567890, Brown,   Palin,      L02, T03, 4, 7.5, 6.5, 0, 5
10012134, Harper,  Ed,         L01, T03, 10, 9, 7.5, 10, 6.5
10014549, Johnson,    Andrew,     L01, T05, 10, 0, 10, 5.5, 7
10020987, Clockwork,  Milan,      L02, T03, 10, 8.5, 8, 9, 9
10021234, Freeman,    Skyski,     L01, T02, 0, 10, 10, 10, 8.5
EOF
```

The first line of the file explains each column of the data. Let n be the total number of students, then the next n lines of the file each corresponds to a student in the class, and contains 10 fields:

(1) Student ID

(2) Last name

(3) First name

(4) Lecture section

(5) Tutorial section

(6-10) Grades for Assignments 1-5

Some fields, such as tutorial section, have fixed length. However, other fields, such as last name, are of variable length. Nonetheless, neighboring fields are always delimited by commas.

Assuming the grades are stored in a file ‘grades.txt’, then you can read an entire line of the file into a Python string s by using the following Python statements:

```
file = open (‘grades.txt’, ‘r’)
s = file.readline()
```

You just need to open the file once, then you can use the readline() function multiple times, to read a successive line each time.

After the n lines of student records, the file ends with a last line that says “EOF”, short for “End of File”.

The number n is not known *a priori.* The sample input that we use to test your program will contain 100 to 300 students, in a file named ‘grades.txt’.

Extracting grades

We wish to eventually draw a histogram for grade distribution of Assignment 1. Therefore you need to extract the grade of A1 for each student, by processing his/her corresponding line in the file.

Hint: your program can traverse the line from the beginning, while counting the number of commas that have been encountered. The slice of the string between the 5th and the 6th commas should be the field for A1 grade. Alternatively, you can do a reverse traversal from the end of the string. That way the string slice between the 4th and the 5th commas should be the field for A1 grade. Or, you can also apply the Python `split` function that splits a long string into a list of shorter strings, using a specified separator.

You may wish to construct a list that will have one entry for each student, storing his/her A1 grade. Each time you extract a new grade, append it to the list.

Obtaining grade distribution

You can assume that each grade is between 0 and 10 (inclusive), and is either an integer or a half-integer, of the form ‘7.5’, ‘4.0’, ‘10.0’, *etc.* In the final histogram, we wish to draw a bar for each grade segment within range $[k, k+1)$, for $k = 0, 1, \ldots, 9$, as well as for a separate segment for grades that are equal to 10. To prepare the data for that purpose, you can construct a second list with 11 entries, where each entry will store the number of grades in a corresponding grade segment.

Visualizing grade distribution

The final step is to visualize the list of segment sizes into a histogram using QuickDraw.

Your Python program should print Quickdraw commands for the following:

- The x axis and the y axis, units and labels for them
- For each grade segment, a bar with a height proportional to the number of grades in that segment.

Note that, upon deciding the unit and range for the y axis, your program needs to consider the number of grades in the segments, such that the highest bar is somewhat close to the top border of the screen yet can still be comfortably accommodated.

(Optional) Computing and displaying other statistics

This part is optional and is not required to obtain the full mark in this assignment. If you wish, try to compute the following statistics of the grades for the class (which you have stored in a list):

- max: the highest grade in the class
- min: the lowest grade in the class

- ave: the average grade for the class
- med: the median grade for the class
- sd: the standard deviation of the grades

Once you have computed these numbers, or some of them, display them to an appropriate location on the screen by printing QuickDraw commands. If you wish to compute a certain parameter listed above, but are not sure what it means or how to compute it, you are welcome to obtain that information through "enquiry based learning", *i.e.*, consulting a reference book, doing Internet search, or asking someone else. The ability to acquire new knowledge and skills by yourself is part of the goal of University training.

Sample input and output

Available at

`http://pages.cpsc.ucalgary.ca/~zongpeng/CPSC231-F12/assignments/A3/a3.html`

Submission

Submit your solution by sending an email to your TA, with your solution program in the attachment. Name your program in the form of `a3-sid.py`. For example, if your student ID is 123456, then name your program `a3-123456.py`.

Submissions received after the deadline will not be accepted.