A201/A597 Introduction to Programming I

First Summer 2007

Lecture Notes for Monday June 11, 2007 (ED1204)

The final exam is coming, we post the problems you need to prepare.

1. Write a function `incrementLetters(s)` that returns a string whose characters are obtained by substituting for each corresponding letter in string `s` the next letter in the alphabet, with `(z, Z)` becoming `(a, A)`. Maintain the same letter case (upper or lower), and leave non-letter characters unchanged. Hints: The built-in functions `ord` and `chr` convert a character to its corresponding integer and viceversa, and the string method `isalpha` is a predicate indicating if its argument is a letter. You may also wish to use the string method `islower` that indiates if its argument is a lower-case letter.

2. Relates to the problem above: Write a function `rotateLetters(n, s)` that has the effect of applying `incrementLetters n` times (that is, replacing each letter by the one `n` characters after it in the alphabet, rotating from (z, Z) back to (a, A)). Try writing this both the easiest way, actually calling `incrementLetters n` times, and more efficiently by iterating over the string only once. Hints for the efficent approach: use the remainder operator, the string method `islower` that indicates the case of a letter, recall that there are 26 letters in the ASCII alphabet, and at first simplify the problem by assuming all characters are upper case.

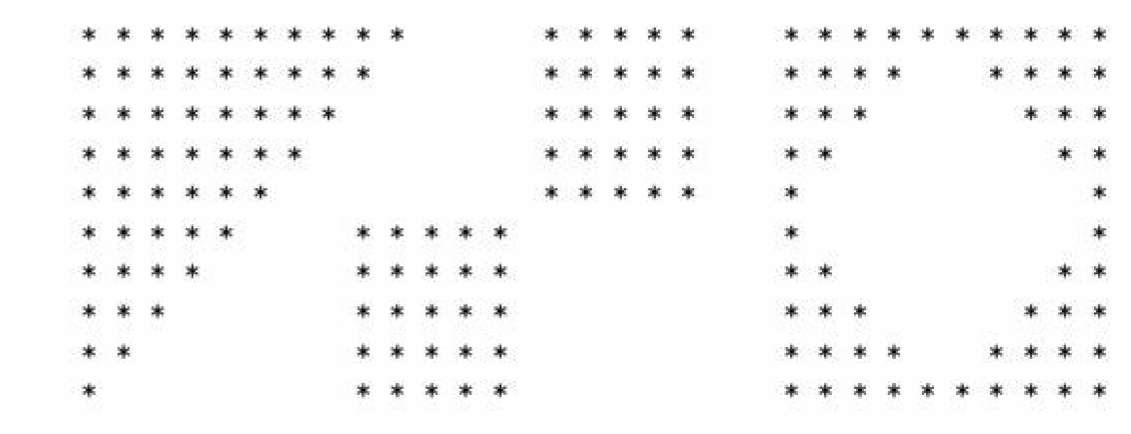
3. Write a function named word_frequency that takes a string, s, and returns a list of tuples associating each word in s with the frequency of its occurence. Consider words to be sequences of characters separated by blank space, making the string method `split` very handy. Word comparison should be case-insensitive, so for example Cat and cat would both contribute to the count the dictionary associates with key cat. Also ignore all characters that are not letters or numbers, so for example end; (including a semicolon) would contribute to the count for the word end. Can you solve this problem without the use of a dictionary?

4. Consider the guess my number game of chapter 3. Write a program in which the player and the computer trade places in the number guessing game. That is, the player picks a random number between 1 and 100 that the computer has to guess. Before you start think about how you guess. That's what the computer program should do it.

5. (Fire Extinguisher) Suppose you determine that the fire extinguisher in your kitchen loses `x` percent of its foam every day. How long before it drops below a certain threshold (`y` percent), at which point it is no longer serviceable? Write a program that lets the user input the `x` and `y` and then reports how many weeks the fire extinguisher will last.

6. `http://www.cs.indiana.edu/classes/a201-dger/spr2005/notes/hwFive.pdf`

7. Write a program for each of the three patterns below. The program should be able to produce that pattern as a scalable pattern (first ask user the size, then produce pattern).



```
* * * * * * * * * *       * * * * *     * * * * * * * * * *
* * * * * * * * *         * * * * *     * * * *     * * * *
* * * * * * * *           * * * * *     * * *         * * *
* * * * * * *             * * * * *     * *             * *
* * * * * *               * * * * *     *                 *
* * * * *       * * * * *               *                 *
* * * *         * * * * *               * *             * *
* * *           * * * * *               * * *         * * *
* *             * * * * *               * * * *     * * * *
*               * * * * *               * * * * * * * * * *
```

.

8. In mathematics, a set is a collection of values in which no value occurs more than once and in which the order of values is not significant. In Python it is natural to represent sets as lists for which the order of elements is not significant and no two elements have equal values.

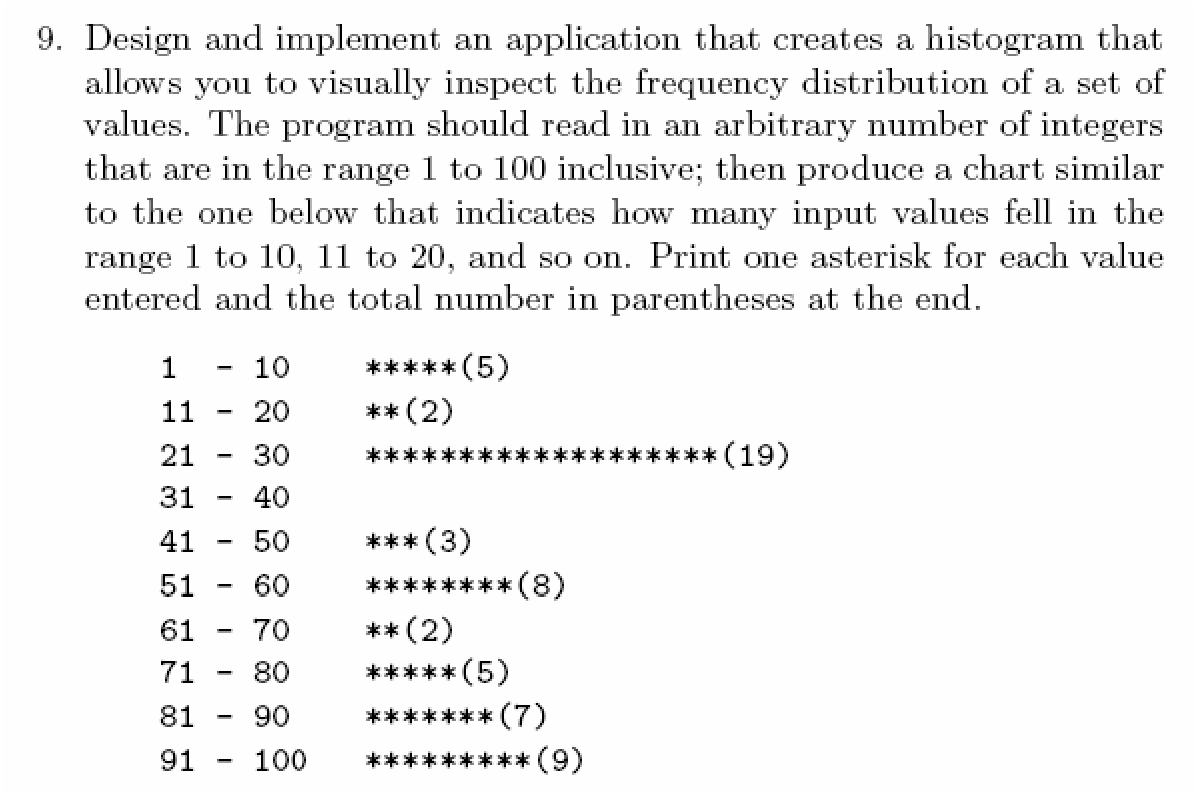
- Write a function named `make_set` that takes a list and returns a set. Hint: copy the list, sort the copy, and remove duplicates from it before returning it.

- Write a predicate named `set_equals` that takes two sets and indicates if they are equivalent as sets (have the same values, but order may be different).

- Write a function named `set_difference` that takes two sets and returns a set of the values in the first set that are not in the second set. (This is related to the difference function described earlier, which was also named after the mathematical set difference operation, but is not the same in several respects.)

- Write a function named `set_union` that takes two sets and returns a set of the values that are in either one.

- Write a function named `set_intersection` that takes two sets and returns a set of the values that are in both sets.



9. Design and implement an application that creates a histogram that allows you to visually inspect the frequency distribution of a set of values. The program should read in an arbitrary number of integers that are in the range 1 to 100 inclusive; then produce a chart similar to the one below that indicates how many input values fell in the range 1 to 10, 11 to 20, and so on. Print one asterisk for each value entered and the total number in parentheses at the end.

```
1  - 10     *****(5)
11 - 20     **(2)
21 - 30     *******************(19)
31 - 40
41 - 50     ***(3)
51 - 60     ********(8)
61 - 70     **(2)
71 - 80     *****(5)
81 - 90     *******(7)
91 - 100    *********(9)
```

10. The wildcard problem from the practical.

Minute paper: to be announced in class.

Essentially they were be parts of the programs above.