

COMP201 — Principles of Programming

Semester I 2009–2010

Quiz #4

Student ID: _____ Name: _____ Score: _____

This is an all-programming quiz.

IMPORTANT: This programming environment is not the same as the one that you are using in class. To save and compile your programs, follow the instructions below:

- Log onto the machine with the following information:
 - Location: ShortCourse
 - Username: **s** + *student id*. For example, if your student ID is 09123456d, log in with the user name **s09123456d**. All lowercase letters.
 - Password: comp201
- Run `nalwin32` and get to Python just like you do in class.
- JEdit 4.2 should be available directly from the **Start** menu, or on your desktop, or from `nalwin32`.
- **Save your programs onto the J: drive. Do not create any subfolders.** We will be using a program to collect your programs, and if you don't save it in the correct place, we won't be able to get it and you will get no marks.

1. (10 marks) For this question, you need to write a program that will:

- Ask the user to type in a word (alphabetical characters only)
- Find the pair of consecutive characters that are closest to each other in the word.

The distance between two characters is defined as the number of characters between them. For example, the distance between “A” and “D” would be 4, because “D” is four characters “down the line” from “A” (we go through “B”, “C”, and then we get “D”).

You can assume that the user-input word is of all lowercase characters, and will be at least 2 characters long. A sample run of your program would look like this:

```
J:\> python closestDistance.py
This program finds the closest pair of characters in a word
Please type in a word: hello
The smallest distance is 0, between characters 2 and 3
J:\> python closestDistance.py
This program finds the closest pair of characters in a word
Please type in a word: university
The smallest distance is 1, between characters 5 and 6
J:\>
```

If there are two pairs with the same minimum distance, report the second of the two pairs.

Some hints for you:

- Your program needs to somehow remember the *current* smallest distance, and the starting position of the current smallest distance. These variables store the partial result (similar to the concept of an accumulator).
- You can use the Python function `abs()` to get the absolute value.
- The Python function `ord()` gets the ASCII value of a character, `chr()` gets the character corresponding to a given ASCII value.

Your program should be called `closestDistance.py`. Leave your program on the J: drive.

For additional challenge:

(1 point extra credit) Modify your program to work with mixed case (both capital and small letter) strings as well as all-lower-case strings. The distance between two characters of different cases is the same as it would be if they were of the same case (in other words, accept mixed-case strings, but ignore case information).

2. (10 marks) The Fibonacci sequence is a series of numbers where the value of each number is equals to the value of the two numbers preceding it. For example, the first 10 Fibonacci numbers are: 1 1 2 3 5 8 13 21 34 55. Note how starting from the 3rd number onwards, each number equals the previous two added together. The first two numbers in the sequence are defined to be equals to one.

Your job is to write a program that will ask the user for the value of *n*, the number of Fibonacci numbers to print out, and print the sequence consisting of those *n* numbers. An example run of your program should look like this:

```
J:\> python fibonacci.py
This program prints out the first n Fibonacci numbers
How many Fibonacci numbers do you want? 8
1 1 2 3 5 8 13 21
```

A hint for you: Treat the first two numbers as “special cases” and deal with them separately. The rest of the values in the series should be processed in the same way, repeatedly.

Your program should be called `fibonacci.py`. Leave it in the J: drive.

-End-