CS 115 – Dubai – Fall 2012

Assignment 02

Due: Thursday, October 4, 2012 at 11:59 p.m.

- For this and all subsequent assignments, you are expected to use the design recipe when writing functions from scratch, including helper functions.
- Do not copy the purpose directly from the assignment description. The purpose should be written in your own words and include reference to the parameter names of your functions.
- The solutions you submit must be entirely your own work. Do not look up either full or partial solutions on the Internet or in printed sources.
- Do not send any code files by email to your instructors or tutors. It will not be accepted by course staff as an assignment submission. Course staff will not debug code emailed to them.
- Test data for all questions will always meet the stated assumptions for consumed values.
- Read each question carefully for restrictions.
- Read the course Web page for more information on assignment policies and how to organize and submit your work. Follow the instructions in the style guide. Specifically, your solutions should be placed in files a02qY.rkt, where Y is a value from 1 to 3.
- Download the interface file from the course Web page.
- For full marks, it is not sufficient to have a correct program. Be sure to follow all the steps of the design recipe, including the definition of constants and helper functions where appropriate.

Language level: Beginning Student
Coverage: Module 3

1. Write a function called `double-string?` that consumes a non-empty string and produces true when the string is formed by two identical strings, one immediately following the other, separated by at most one character. For example `"bye-bye"` and `"yoyo"` are double strings, but `"good"`, `"edited"`, and `"level"` are not double strings. You may use functions (or modified versions of them) from your course notes to help solve this problem. However, if you use those functions, you must include acknowledgement of where they came from in your comments.

2. You have decided to take a walking tour of Bastakiya on the weekend. There are three tours available: the one-hour tour for 35 AED, the 90-minute tour for 55 AED, or the two-hour tour, which includes lunch, for 255 AED per person. Also, if you arrange to get 10 people or more as a group, you can get a 10% discount on the cost per person. For example if you have 12 people who want to take the 90-minute tour, then each would be charged 49.5 AED.

   Write a function called `tour-total` that consumes two values: `num-people` and `tour-type`. The parameter `num-people` is a natural number indicating the number of people in your group. The parameter `tour-type` will be one of `'one-hour`, `'90-min`, or `'two-hour`, indicating which tour your group wants to take. The function should produce a number that is the total cost for all participants in the walking tour. For example: if 12 people want to take the 90-minute tour, the total cost would be `594`.

   *** It is very important that when you check symbols that you are using the exact values described in this question. For example `'one-hour` is not the same as `'One-hour`. If you are not exact, you could lose many correctness marks.

Assignment 02

Due: Thursday, October 4, 2012 at 11:59 p.m.

3. All data in a computer is stored as a sequence of 0's and 1's known as bits. The sequence can be referred to as a binary number. Sometimes bit errors may occur when transmitting a binary number. One way to detect errors is by checking the parity of the number. The parity of a binary number can be even or odd. If it is even, that means there is an even number of 1's in the binary number. If it is odd, that means there is an odd number of 1's in the binary number. If you know that the parity is supposed to be even or odd, then you can check the bits in the number to make sure they match the requirement.

   Write a function called `bit-error?` that consumes two values `bin-num` and `parity`. The parameter `bin-num` is a string of length four that contains only 0's and 1's. The parameter `parity` will be `true` for even parity and `false` for odd parity. The function produces `true` when the `bin-num` does not match the parity and `false` otherwise. For example, `(bit-error "1001" false)` produces `true`.

   A solution that simply checks each of the 16 possible binary numbers and produces true or false accordingly, will not receive full correctness marks.