CS 420 Advanced Programming Languages
Fall Semester, 2022
Assignment 3
San Diego State University -- This page last updated on 8/24/22

Assignment 3
Due Oct 9 at 11:59 pm

Use Clojure to implement the following. Each problem is worth 5 points.

1. We can use a map to describe an item on a restaurant bill. For example {:name “Green Tea Ice Cream” :price 2.5 :quantity 2}. We can represent the bill as a vector of maps. Write a Clojure function **bill-total** whose one argument is a vector of such maps and returns the total of the bill. For example

   (def bill [{:name “Green Tea Ice Cream” :price 2.5 :quantity 2}
   {:price 1.0 :name “Sticky Rice” :quantity 1}])
   (bill-total bill) returns 6.0

2. Often people will order additional times. Write a function **add-to-bill** that accepts two arguments. The first is a bill as above. The second argument is a vector of additional items. The method returns a new bill with the additional items. For example:

   (def items [{:price 2.1 :name “Mango” :quantity 1}{ :quantity 1:price 1.0 :name “Sticky Rice” }

   (add-to-bill bill items) returns [{:name “Green Tea Ice Cream” :price 2.5 :quantity 2}
   {:price 1.0 :name “Sticky Rice” :quantity 2}
   {:price 2.1 :name “Mango” :quantity 1}]

3. Write a function, divisors, with one argument, a positive integer. The function returns a sequence of the divisors of N.

4. An abundant number is an integer for which the sum of its proper divisors is greater than twice the number. For example, 12 is an abundant number as its divisors are 1, 2, 3, 4, 6, and 12, totaling 28. Write a function, abundance, that has one argument, an integer, and returns the sum of the proper divisors of the number minus the number itself. For example, (abundance 12) returns 4.

5. Find all the abundant numbers less than 300.

6. Write a function, pattern-count, with two arguments. The first argument is a string, let's call it text, and the second argument is also a string; call it a pattern. The function pattern count returns the number of times the pattern occurs in the text. For example

   (pattern-count “abababa” “aba”) returns 3
   (pattern-count “aaaaa” “aa”) returns 4
   (pattern-count “Abcde” “abc”) returns 0

Answer all questions in a single Clojure file (file extension .clj). Use a comment to separate and label each question. Place the questions in order in your file.

Late Policy

An assignment turned in 1-7 days late will lose 5% of the total value of the assignment per day late. The eight-day late penalty will be 40% of the assignment, the ninth-day late penalty will be 60%, and after the ninth day late, the penalty will be 90%. Once a solution to an assignment has been posted or discussed in class, the assignment will no longer be accepted. Late penalties are always rounded up to the next integer value.