CS 596 Functional Programming & Design
Spring Semester, 2014
Assignment 3

San Diego State University -- This page last updated 9/12/14

Due Sept 23 23:59

You may find it useful to use more than one function to solve each problem.

1. Write a function called sdsu-rotate which can rotate a sequence in either direction. The function can return a collection type of your choice. Here are some examples.

   (sdsu-rotate 2 [1 2 3 4 5]) returns (3 4 5 1 2)
   (sdsu-rotate -2 [1 2 3 4 5]) returns (4 5 1 2 3)
   (sdsu-rotate 6 [1 2 3 4 5]) returns (2 3 4 5 1)

2. If we list all the positive integers below 10 that are multiples of 2 or 5, we get 2, 4, 6, 8, 5. The sum of these multiples is 25. Write a function sdsu-sum that takes three arguments. The first two arguments are the numbers we will take the multiples of. The third argument the the number we want the multiples to be less than. Sdsu-sum returns the sum of the multiples of the two numbers less than the third argument. For example:

   (sdsu-sum 2 5 10) returns 25

3. A palindromic integer is an integer that reads the same both ways. For example, 101, 121, 12321 and 98989 are all palindromic integers. The largest palindromic integer made from the product of two 2-digit numbers is 9009 = 91 × 99. Write a function sdsu-palindome that takes one argument, an integer. The integer indicates the number of digits we are interesting in. So (sdsu-palindrome 2) returns the largest palindromic integer made from the product of two 2-digit numbers and (sdsu-palindrome 3) returns the largest palindromic integer made from the product of two 3-digit numbers.

4. The Fibonacci sequence is generated by adding the previous two terms in the sequence. We will start the sequence with 1 and 2. So the first ten elements are 1, 2, 3, 5, 8, 13, 21, 34, 55, 89. Write a function sdsu-fibonacci-even that takes one argument N. The function returns the sum of all the even-valued terms in the Fibonacci sequence that are less then or equal to N.

5. DNA can be represented as a string containing the characters A, C, G and T. These letters represent the four nucleobases adenine (A), cytosine (C), guanine (G), and thymine (T). An example of a DNA string is "ATGCTTCAGAAAGGTCTTACG". Write a functions sdsu-dna-count that takes one argument, a DNA string, and returns a map that indicate the number of times each of the four characters appears in the DNA string. So for example we have:
   (sdsu-dna-count "ATGCTTC") returns {:A 1 :T 3 :G 1 :C 2}
   (sdsu-dna-count "ATGCTTCAGAAAGGTCTTACG") returns {:A 6 :T 6 :G 5 :C 4}

6. Write a function, sdsu-digits, with two arguments, n and b. The first argument, n, is a non-negative number in base 10. The second argument, b, is a positive integer also in based ten. Sdsu-digits converts the first number from base 10 to base b. The return value is a collection of the digits not as one number or as a single string. Digits should be represented with their integer values. For 15 would be [1 5] in base 10, [1 1 1 1] in base 2 and [15] in base 16. So for example we have:

   (sdsu-digits 12348012 10) returns [1 2 3 4 8 0 1 2]
   (sdsu-digits 0 9) returns [0]
   (sdsu-digits 140 8) returns [2 1 4]
   (sdsu-digits 110 16) returns [6 14]

7. Write a function sdsu-roman-numeral has one argument, an integer smaller than 4000. The function returns the corresponding roman numeral in uppercase. See http://en.wikipedia.org/wiki/Roman_numerals for information about roman numerals.

   (sdsu-roman-numeral 1) returns "I"
   (sdsu-roman-numeral 30) returns "XXX"
   (sdsu-roman-numeral 4) returns "IV"
   (sdsu-roman-numeral 9) returns "IX"
   (sdsu-roman-numeral 40) returns "XL"
   (sdsu-roman-numeral 90) returns "XC"
   (sdsu-roman-numeral 400) returns "CD"
   (sdsu-roman-numeral 900) returns "CM"
   (sdsu-roman-numeral 1904) returns "MCMIV"

What to Turn in

Create one file with all the problems answered. Make sure that the function names are as indicated in the problems. Zip the file up and upload the zipped file to assignment two in the course portal.

Late Penalty

An assignment turned in 1-7 days late, will lose 3% of the total value of the assignment per day late. The eight day late the penalty will be 40% of the assignment, the ninth day late the penalty will be 60%, 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.