Universität Bonn
Institute of Computer Science II
Programming Lab I

Prof. Dr. Andreas Weber
Mohammad Khatami
Christoph Lüders

Exercise sheet 3

27.10.2016

This exercise sheet is due 15:45 on Thursday, 10.11.2016. Please return the solved exercises in class as a printout of your source code plus the output. Put your name and email address on the printout.

Use comments and meaningful variable and function names. Use docstrings for functions. Use `assert` where it is useful. Choose *meaningful* input values for demonstration, that is, ensure that every part of your code is run, especially the corner cases. Don't forget to test your code!

Exercise 1: Complement a DNA string

Write a function that takes as input a strand of DNA, in a `string` over the alphabet "ACGT" and returns its complemented DNA string. "A" and "T" are complements of one another, as are "C" and "G". Example: the complement of "ACATGAGA" is "TGTACTCT". Raise an exception if input characters are not part of the alphabet "ACGT". Provide at least five inputs and their return values and handle any exceptions. *(6 pt)*

Exercise 2: Multiply two polynomials

A polynomial (in x) over $\mathbb{R}$ is the sum $\sum_{i=0}^{n} a_i x^i$, with $a_n \neq 0$. Write a function that takes as input two polynomials and returns their product. A polynomial is represented as a `list`, where the entry at index i corresponds to the coefficient a_i. For example, the polynomial $5x^3 - x + 14.2$ is represented by `[14.2, -1, 0, 5]`. Make sure that the leading coefficient of the result is non-zero. Show at least five inputs and their return values. *(6 pt)*

Exercise 3: ∗ Anagram checker

An *anagram* is a new word or phrase that consists of all the letters of a given original word or phrase, but in different order. Only letters are considered, spacing and any punctuation is ignored, as is the case of the letters. Here are some example anagrams:

- William Shakespeare ≃ I am a weakish speller,
- Tom Marvolo Riddle ≃ I am Lord Voldemort,
- Lab Course: Programming Lab ≃ Blamable program sourcing.

Write a function that takes two `string`s as input and returns `True` if they are anagrams of each other and `False` otherwise. Provide at least five pairs of strings of your own choosing with their return values, where all but two are anagrams. *(∗ 4 pt)*

Mailing list

There is a mailing list for students of the LSI programming lab course. The web page is:

https://lists.iai.uni-bonn.de/mailman/listinfo.cgi/lab-lsi1

This mailing list is to exchange information on the course and exercises. Please give only hints, but no solutions!