

Programming Paradigms

Caucasus University – Spring 2022

Assignment 5 (10 points)

– Deadline: **23:59** on Friday 23 Dec 2022 –

1 Sequence (2 points)

Write a function *sequence* that takes 3 arguments *low*, *high*, and *stride*, all assumed to be numbers. Further assume stride is positive. *sequence* produces a list of numbers from *low* to *high* (including *low* and possibly *high*) separated by *stride* and in sorted order.

Examples:

```
> (sequence 3 11 2)
' (3 5 7 9 11)
> (sequence 3 8 3)
' (3 6)
> (sequence 3 2 1)
' ()
```

2 String-append-map (2 points)

Write a function *string-append-map* that takes a list of strings *xs* and a string *suffix* and returns a list of strings. Each element of the output should be the corresponding element of the input appended with *suffix* (with no extra space between the element and suffix). You must use Racket-library functions *map* and *string-append*.

3 Sum-Square Difference (2 points)

The sum of the squares of the first ten natural numbers is

$$1^2 + 2^2 + \dots + 10^2 = 385$$

The square of the sum of the first ten natural numbers is

$$(1 + 2 + \dots + 10)^2 = 3025$$

Hence the difference between the square of the sum of the first ten natural numbers and the sum of the squares is $3025 - 385 = 2640$.

Write a function *sum square diff n* that returns the difference between the square of the sum and the sum of the squares of all natural numbers from 1 to *n*.

You must decide how to handle the case when $n < 1$.

4 Counting DNA Nucleotides (2 points)

<http://rosalind.info/problems/dna/>

A *string* is simply an ordered collection of symbols selected from some *alphabet* and formed into a word; the *length* of a string is the number of symbols that it contains.

An example of a length 21 *DNA string* (whose alphabet contains the symbols 'A', 'C', 'G', and 'T') is "ATGCTTCAGAAAGGTCTTACG."

Given: A DNA string *s* of length at most 1000 nt.

Return: Four integers (separated by spaces) counting the respective number of times that the symbols 'A', 'C', 'G', and 'T' occur in *s*.

example:

```
> (count-nucleotides "AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATT
    AAAAAAAGAGTGTCTGATAGCAGC")
20 12 17 21
```

5 Transcribing DNA into RNA (2 points)

<http://rosalind.info/problems/rna/>

An *RNA string* is a string formed from the *alphabet* containing 'A', 'C', 'G', and 'U'.

Given a DNA string *t* corresponding to a coding strand, its transcribed RNA string *u* is formed by replacing all occurrences of 'T' in *t* with 'U' in *u*.

Given: A DNA string *t* having length at most 1000 nt.

Return: The transcribed RNA string of *t*.

example:

```
> (dna-to-rna "GATGGAAGTGGACTACGTAAATT")
"GAUGGAACUUGACUACGUAAAUU"
```

Submitting

Zip all source code and upload it on teams.

Grading

Correctness: 80% Your code should compile without any errors or warnings. We will test your programs on different inputs and edge cases. For a full score your code should pass all tests (some exercises might already have predefined tests).

Code Quality: 20% We expect your code to be clean and readable. We will look for descriptive names, defined constants (not magic numbers!), and consistent layout. Be sure to use the most clear and direct C syntax and constructs available to you.