Programming languages – Haskell

Homework exercise (2021/22)

T. Goluch

1st exercise (1 pts.)

For a given number **n** and a list **L** of numbers, find the sum of all numbers from **1** to **n** divisible by at least one number from **L**.

Comments:

a. all functions should have an appropriate header with the type of function,
b. in programs you cannot use functions outside of this instructions,
c. the sort function is only allowed in the task below.

2nd exercise (6 pts.)

Write a program to solve propositional calculus[1] with: negation (N), conjunction (C), alternative (A) and implication "conditional" (I) functors.
Sentence type: data Sentence = S Char | … - i.e. the variable can be any variable S
Example sentence: `(I (N (S 'p')) (A (C (S 'p') (S 'q')) (S 'r')))`

Program should:

1) contains a function: `print sentence` – (1.5 pts.)
for a given example `(I (N (S 'p')) (A (C (S 'p') (S 'q')) (S 'r')))`
the output should be `"(~p => ((p & q) | r))"`

2) contains a function: `write_variables sentence` – (1.5 pts.)
for a given example `(I (N (S 'p')) (A (C (S 'p') (S 'q')) (S 'r')))`
the output should be `[p, q, r]` (not necessarily sorted, but unique)

3) contains a function: `check sentence values_map` – (1.5 pts.)
for a given example `(I (N (S 'p')) (A (C (S 'p') (S 'q')) (S 'r')))`
and values map `fromList [('p', False), ('q', True), ('r', False)]`
the output should be `False`

4) contains one of the functionalities a/b – (1.5 pts.)
4a) perform tasks 1-3 for Łukasiewicz's three-valued logic[2] `True`/`False`/`Nothing`
4b) contains a function: `is_tautology` checking whether the sentence is true for any valuation of variables in the formula.

[1] https://en.wikipedia.org/wiki/Propositional_calculus
[2] https://en.wikipedia.org/wiki/Three-valued_logic