Instructions for Students

- Write your name and student number on the exam sheet and on every solution sheet you hand in and also sign them.
- This is a closed book exam: the only resources allowed are blank paper and pens (do not use pencils).
- Write neatly and clearly. The clarity of your explanations will affect your grade.
- The duration of the exam is 2 hours.

Good luck!

Do not write in this space

<table>
<thead>
<tr>
<th>Question</th>
<th>Marks</th>
<th>Achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Exercise 1 (24 marks)

a. (4 marks) Briefly describe the concept of tail recursion, and why it is desirable to write tail recursive functions.

b. (4 marks) Given is the expression [20, "20.0", 20.0].
   - Is the expression legal in Ruby?
   - Is the expression legal in Haskell?

   Explain your answers and briefly describe the meaning of this expression.

c. (4 marks) Briefly describe the difference between the following expressions in Prolog:
   - ?- 7 = 3 + 4
   - ?- 7 is 3 + 4

   What will be the result of executing these expressions?

d. (4 marks) What is wrong with the following case statement in Erlang? How could you fix the code?

   ```erlang
   case X of
     {_,_} -> doA;
     {_,3} -> doB;
     {2,_} -> doC;
     {2,3} -> doD
   end.
   ```

e. (4 marks) Briefly explain the concept of list comprehension in Haskell and give a short example.

f. (4 marks) Consider the Haskell function `prod x y = x * y`. How is the function call `prod 2 4` evaluated? What is the name of this evaluation concept?

Exercise 2 (12 marks) Assume temperature data stored in an array `t`. Write a Ruby function `maxperiod(t,x)` that calculates the length of the longest (hot) period with temperature values greater than `x`. The result is printed to the console. For example, for `t = [20, 25, 26, 23, 27]` and `x = 24`, the function prints  The longest period greater than 24 is of length 2.

Exercise 3 (8 marks) Write a Ruby function `palindrome(a)` to check whether an array of characters represents a palindrome, i.e., is identical to the reversed array. For instance, the array `['A', 'B', 'B', 'A']` represents a palindrome, while `['A', 'B', 'B', 'A', 'C']` does not. You are not allowed to use Ruby’s `array` method `reverse`.
Exercise 4 (10 marks) The product of two natural numbers can be expressed as a repeated addition using the following recursive definition (Peano axioms):

\[
0 \times y = 0 \\
x \times y = (x - 1) \times y + y
\]

Write a Prolog program to compute a product using this definition.

Exercise 5 (10 marks) Write a Prolog program subst(X,L1,Y,L2) that replaces all occurrences of X in list L1 by two occurrences of Y to obtain list L2. For example, subst(a,[a,b,c],1,L2) instantiates L2=[1,1,b,c].

Exercise 6 (10 marks) Write a function loop for an Erlang process that receives messages consisting of a single parameter and does the following:

- if the parameter is a number, it outputs to the console whether the number is positive, negative, or zero;
- if the parameter is “bye”, the process terminates;
- otherwise, an error message is printed, e.g., “Unexpected message”.

Show also how to start the process. (Hint: you can use a function is_number(N), which is true if N is a number, and false otherwise)

Exercise 7 (12 marks) Write a Haskell function diffrev, which takes as input two lists, A and B, each containing n numbers, and produces in output a list C such that

\[
C[0] = A[0] - B[n-1] \\
\vdots \\
C[n-1] = A[n-1] - B[0]
\]

For example, diffrev [1,2,10] [2,5,-2] returns [3,-3,8]. (Hint: think about breaking the problem down in more than one function!)

Exercise 8 (14 marks) Write a Haskell module that exports a tail-recursive function noOfElem that takes as input an element x and a list and returns the number of occurrences of x in the list. For instance, noOfElem 1 [1,2,3,1] returns 2.