Task 1.
Consider the following imperative pseudo code for computing the factorial function:

procedure fact
  input n;
  k = 1;
  while n > 1 do
    k = n * k;
    n = n - 1;
  od
return k

Write a corresponding tail-recursive Haskell-program.
Explain why your program is tail-recursive.
Hint: Study the example of reversing lists in the lecture slides.

Task 2.
Define a function that takes a list of integers and computes the list of its absolute values (you may use the predefined function abs).

Define this function in two ways:
1. Using list comprehension.
2. Using map

Task 3.
Define a function that takes a list of integers and computes the list of those elements of the list that are even (you may use the predefined function even).

Define this function in two ways:
1. Using list comprehension.
2. Using filter

Note:
Lab tasks have to be performed on the date shown during the lab session and seen by a lab supervisor.