

Problem Set 5

Semester 1, 2011/12

Due: October 13, 23:59

Marks: 6

Submission: In IVLE, in the cs2104 workbin, you will find a folder called “Homework submissions”. In that folder, there are currently 3 *subfolders*: **PS5P01**, **PS5P02**, and **PS5P03**. The last two digits of the folder name indicate the solution that is to be submitted into that folder: the solution to *Question 1* into **PS4P01**, and so on (that is, you need to submit 3 separate solutions to 3 problems). A solution should consist of a *single text file* that can be compiled, or loaded into the interpreter of interest and executed. You should provide as much supplementary information about your solution as you can, *in the form of program comments*. Moreover, if you work in a team, state the members of the team at the beginning of the file, in a comment. You do not need to submit the same file twice, one submission per team is sufficient.

Important note: In this problem set, you will practice the use of the higher order functions `map` and `foldr` in building concise and expressive programs. As such, each problem comes with a *solution template*, which indicates angle-bracketed `<code placeholders>` that are to be filled in to complete the solution. You are only allowed to add code inside those placeholders. The rest of the template must remain unmodified.

Problem 1 [2 marks, submit to PS5P01]

Write a Haskell function that inserts an element at a given position in a list. The position should be counted from the end of the list. Use the following code template:

```
insert a pos l =  
  fst ( foldr  
        (\x (b,c) -> <your code here> )  
        ((drop pos <your code here> ),0) l )
```

The following are usage examples for `insert`. Your implementation must replicate them accordingly:

```
> insert 10 0 [1,2,3]
[1,2,3,10]
> insert 10 1 [1,2,3]
[1,2,10,3]
> insert 10 5 [1,2,3]
[1,2,3]
> insert 10 0 []
[10]
> insert 10 1 []
[]
> insert 10 1 [0]
[10,0]
```

Problem 2 [2 marks, submit to PS5P02]

Write a higher-order Haskell function that counts the number of occurrences of a number x in a list l . Use the following code template:

```
count x l = foldr <your code here> 0 <your code here>
```

Problem 3 [2 marks, submit to PS5P03]

Write a C version of the half-interval method presented in Lecture 7. Instead of a function argument for the higher-order procedure `solve`, use a pointer-to-function argument. Add a `main` function to your program, and show that your implementation is correct by means of several tests.

Further Practice Problems

These problems are for your own individual practice. Solutions are not to be submitted, and will not be marked. You are, however, allowed to post your solutions in the forum for comparison and discussion. Good posts will earn marks.

Further Practice Problem 1

Further Practice Problem 2