159.202 Assignment 5

Deadline:	**Anytime before** Sunday 11 Oct 2015, time: mid night
Evaluation:	10 marks – which is 3% of your final grade
Late Submission:	5% per hour (or fraction of hour) it is late
Team:	The assignment can be done individually or in pairs.
Purpose:	Practice with lists and functions

This assignment consists of three exercises. You are expected to submit a single file named *a5.hs* containing

- your name(s) and ID(s) (as comments at the top of the file),
- solutions for all exercises, include type definition for every function/value.

Test your final version of *a5.hs* files and make sure it compiles and has no syntax or logical errors.

Exercise 1 [3 marks]

a) Define a recursive function hasD which takes a non-negative integer n and an integer digit d between 0 and 9 inclusive, and returns True if the base-10 representation of n contains the digit d; False otherwise. For example, hasD 3071 7 should evaluate to True, and hasD 1201 4 to False.

b) Define a recursive function `count :: [ Int ] -> Int -> Int` which, given a list of non-negative integers `intL` and an integer digit `a` between 0 and 9 inclusive, returns the number of integers in `intL` which contain the digit `a` (again when written in base 10). For example, `count [102,41,256] 2` should evaluate to 2, and `count [98,4,376] 2` to 0.

c) Re-write the function from point b define using list comprehension-call it `countC`.

Exercise 2 [4 marks]

a) Write a recursive function `takeOut::Int->[Int]->[Int]` so that `takeOut m xs` removes all multiples of `m` from `xs`.

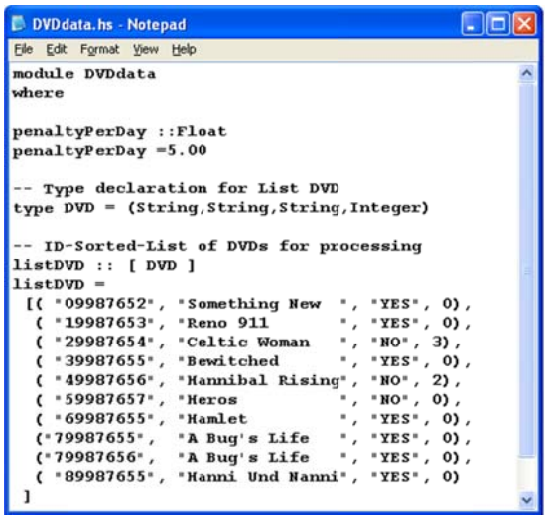
b) Re-write the function from point a using list comptrehension-call it `takeOutB`.

c) Write a recursive function called `count3r` that takes a list of triples of `Int`s as a parameter. The result should be the number of triples for which the last element is the sum of the first two elements. For example:
`count3r [(6,-8,-2),( 2,4,6),(8,7,1),(4,5,9)]` should return `3`
and `count3r [(-1,3,1), (1,5,7)]` should return `0`

d) Re-write the function from point c) using list comprehension;call it `count3L`

Exercise 3 [3 marks] Write a Haskell script that will display some information about data stored in a file called *DVDdata.hs* The file *DVDdata.hs* contains a list of DVD movies and for each movie it contains its ID number, its title, whether it is available

(yes, no) and if it is overdue, how many days it is overdue. The list is sorted according to the movie ID values.
Here is a possible *DVDdata.hs* file:



Your script should print a table with
a) proper headings and
b) containing the following information for each DVD:

i. ID number
ii. title
iii. availability
iv) if overdue – the required payment . The payment for overdue DVDs is computed by multiplying the number of overdue days by `penaltyPerDay`.

c) a footer containing the total number of available DVDs.
d) The list of DVDs should be now sorted on the title of the DVDs.
e) The output should be produced when the function printable (see picture below) is invoked at the prompt.

The output obtained from the above file should be like this:

```
Ok, modules loaded: DVDdata, Main.
*Main> printTable listDVD

==$&$&$&  Welcome to the DVD Hoopy Doo Shop  $&$&$&==

DVD id       Title            Available        Payment

79987661     A Bug's Life          YES
79987659     A Bug's Life          YES
39987655     Bewitched             YES
29987654     Celtic Woman          NO       $15.0
69987658     Hamlet                YES
89987665     Hanni Und Nanni       YES
49987656     Hannibal Rising       NO       $10.0
59987657     Heros                 NO
19987653     Reno 911              YES
09987652     Something New         YES

             DVDs in stock:   7
*Main>
```

Be aware that the content of the file *DVDdata.hs* can be changed (for example more DVDs can be included in the list), but the format of the information will be preserved.

If you have any questions about this assignment, please ask the lecturer before its due time!

Submit your solution electronically using 159.202 Stream.

Important:

1. You can define and use other functions in order to solve a problem that asks you to define a function to perform a specific task.
2. The assignment can be done individually or in teams of at most 2 students-**send one solution file per team**. All assignments authored by 3 or more students will get 0 marks.
3. Different teams submitting suspicious similar solutions will get all 0 marks.
4. Run your final version of *a5.hs* file on Albany computer labs and make sure it has no errors. **Please note that if we cannot run your a5.hs script you will get 0 marks.**