The University of Melbourne

School of Computing and Information Systems

COMP90048 Declarative Programming

Sample mid-semester test

Semester 1, 2018

Write your enrolment number here:

Reading Time: Five (5) minutes.

Writing Time: Forty-five (45) minutes.

This paper has 6 pages including this cover page.

Authorized Materials: No materials are authorized. Calculators are *not* permitted.

Instructions to Invigilators: Students will write all of their answers on this exam paper. *Students may not remove any part of the paper from the examination room.*

Instructions to Students:

- Start by filling in the box above with your student number. Unidentified test papers yield no marks for anyone.
- This test counts for 10% of your final grade. *All questions should be answered* in the boxes provided on the paper. Only material written inside the boxes will be marked.
- Answers should be kept short and should display good programming style.
- The reverse side of any page may be used to make rough notes, or prepare draft answers.
- Unreadable answers will be deemed wrong.
- Use a blue or black pen or pencil.
- You do not need to write comments in your code, but you may include comments if you feel that they would assist the examiner in understanding your code.

Library: This paper may *not* be held by the Baillieu Library.

Question	1	2	3	4	5	Total
Marks						

Question 1 (2 marks)

For each of the following Haskell expressions, write down its **type** (which may be a function type or may include type class constraints) or say that it represents a type error.

A `"abc" ++ "def"`

B `(+)`

C `map length`

D `length : "xyz"`

E `[1, 2, 3]`

Question 2 (1 mark)

Write a one-sentence Haskell comment explaining the purpose of the following function.

```
q2 _ _ [] = []
q2 a b (c:cs) =
    if a == c then b:(q2 a b cs)
    else c:(q2 a b cs)
```

Question 3 (1 mark)

Write a one-sentence Haskell comment explaining the purpose of the following function.

```
q3 a (b:[]) = a b
q3 a (b1:b2:bs) = q3 a (b2:bs)
```

Question 4 (3 marks)

Given the following data type for representing HTML

```
type HTML = [HTML_element]
data HTML_element
    = HTML_text String
    | HTML_font Font_tag HTML
    | HTML_p HTML
    | HTML_ul [HTML]
    | HTML_ol [HTML]
data Font_tag = ...
```

write a Haskell function

```
strip_font_tags :: HTML -> HTML
```

which removes all font tags (replacing them with their HTML contents).

Question 5 (3 marks)

Multi-way trees, trees in which a node may have an arbitrary number of children, can be represented by a type like this, which puts all the children of a node into a list:

```
data Mtree a = Mnode a [Mtree a]
```

Write a function

```
showMtree :: Show a => Mtree a -> String
```

which returns a multi-line string depicting an Mtree. Each node should be placed on a separate line. You can include `\n` in the string for a newline. The children of a node should be placed on subsequent lines, indented by one more space than the line giving the value in the node. For example, given the tree

```
Mnode 1 [Mnode 2 [], Mnode 3 [Mnode 4 []]]
```

the output string should be

```
1
 2
 3
  4
```

Overflow answers

If you do need to use this page, indicate **CLEARLY** in your previous answer that you have continued onto this page. Without such an indication, it is possible that this part of your answer will be overlooked.