COMP313-08A Programming Languages

Second Haskell coursework

1. Write a function

```
halve :: [a] -> ([a],[a])
```

that splits an even-length list it two equal-length halves, so:

```
halve [1,2,3,4,5,6] = ([1,2,3],[4,5,6])
```

2. Define a function

```
mix :: [Int] -> [Int] -> [Int]
```

which mixes two *sorted* lists of integers together, to get a sorted list which contains all the elements from the two original lists, so:

```
mix [3,6,9] [1,8,10] = [1,3,6,8,9,10]
```

The definition should use explicit recursion and no other pre-defined sorting functions.

3. Use the function `mix` from question two to define a function

```
sortints :: [Int] -> [Int]
```

which sorts a list of integers. The empty list and a singleton list are already sorted, and any other list should be sorted by mixing together the two lists that result from sorting the two halves of the list separately.

5. Do exercise 7.5 from the text book.

6. Do exercise 8.1 from the text book.

7. Do exercise 8.3 from the text book.

8, Do exercise 9.2 from the text book.

9. Do exercise 9.9 from the text book.

10. Do exercise 9.10 from the text book.

11. Do exercise 9.11 from the text book.

12. Do exercise 11.1 from the text book.

Your answers for all the questions above are due at 1000 on Wednesday 16th April 2008.
You must submit your answers as a plain text file via Moodle. This **MUST** be a plain text file (**not** a PDF, **not** a MS-format file or any other sort of particular format) since we will want to load your solutions and try them out in hugs. Please also put your name and number at the top of the file.