Question 1 [10 marks]

(a) David wants a function `del1` that takes a Boolean function and a list and returns the list with the first element for which the function is true removed. So if the list is `[1,2,3,4,5]` and the condition is `(> 2)`, the resulting list should be `[1,2,4,5]`. The code must be able to recognize the case where the function is false for all elements, and then should not return a list (because that would be wrong, as one element cannot be removed). It must not give a runtime error. You may assume any functions and types on the last page of the exam.

i) [2 marks] Give an appropriate type for `del1`

ii) [4 marks] Give an implementation for `del1`. Feel free to use auxiliary functions.

(b) [4 marks] Write a function `del1all` that takes a Boolean function and a list and returns the list of all of the lists with exactly one element for which the function is true deleted. It should have the following behaviour (perhaps with the elements of the list in a different order):

```
Prelude> del1all (=='a') "avatar"
["vatar","avtar","avatars"]
Prelude> del1all (=='b') "avatar"
[]
```

Question 2 [12 marks]

This function is supposed to take a list and check whether or not it is a sorted list:

```
sorted :: [a] -> Bool
sorted [] = True
sorted (x:y:xs) = x <= y && (sorted y:xs)
```

This was put into the file `sorted.hs`. As it is, the function has two compile time and one run time error. The line number and the first line of the error message are given below. When answering each subquestion, assume the bugs in previous subquestions have been fixed. For each case, briefly explain why the occurs and propose a fix.

(a) [4 marks]

```
sorted.hs:3:30: error:
    Couldn’t match expected type ’Bool’ with actual type ’[Bool]’

Why error occurred:
Fix:
```

(b) [4 marks]

```
sorted.hs:3:19: error:
    No instance for (Ord a) arising from a use of ’<=’

Why error occurred:
Fix:
```
(c) [4 marks]

Main> sorted [1,2,3,4]
*** Exception: sorted.hs:(2,1)-(3,42): Non-exhaustive patterns in function sorted

Why error occurred:
Fix:

Question 3 [15 marks]

Consider the declaration:

data Expr q a = Ret a
    | Myif (q -> Bool) (Expr q a) (Expr q a)
instance (Show a) => Show (Expr q a) where
    show (Ret a) = show a
    show (Myif _ a1 a2) = "("++show a1++"; "++show a2++")"

eval _ (Ret a) = a
eval a (Myif f t1 t2)
    | f a = eval a t1
    | otherwise = eval a t2

tt = Myif (>5) (Ret "aaa") (Myif even (Ret "bbb") (Ret "ccc"))

(a) [2 marks] What do q and a represent in the first line?
(b) [2 marks] What is the inferred type of eval?
(c) [3 marks] Explain in English (suitable for one of your peers who is just starting to learn Haskell) the point of the line that starts with instance and the two lines that follows. (Do not try to explain the meaning of what is on the right of =. To get full marks you need to explain both the part on the left of => and the part on the right.)
(d) [2 marks] Why is it appropriate to not also have “Show q” on the left side of the =>? Question 3 (cont.) The code from the previous page:

data Expr q a = Ret a
    | Myif (q -> Bool) (Expr q a) (Expr q a)
instance (Show a) => Show (Expr q a) where
    show (Ret a) = show a
    show (Myif _ a1 a2) = "("++show a1++"; "++show a2++")"

eval _ (Ret a) = a
eval a (Myif f t1 t2)
    | f a = eval a t1
    | otherwise = eval a t2

tt = Myif (>5) (Ret "aaa") (Myif even (Ret "bbb") (Ret "ccc"))

(e) [1 mark] What is the value of
eval 3 tt

(f) [1 mark] Suppose the file has been loaded into ghci. What is printed in ghci when the following is typed?

tt

(g) [4 marks] Define the function (including the type declaration) \(\texttt{vals expr}\) that takes an expression \(\texttt{expr}\) and returns the list of values \(x\) such that \(\texttt{Ret x}\) is in \(\texttt{expr}\).

\begin{verbatim}
Prelude> vals tt
["aaa","bbb","ccc"]
\end{verbatim}

Question 4 [10 marks]

Consider representing \(9 \times 9\) matrices as functions from indices to values. That is, \(m_{ij}\) is the \((i,j)\)-th element of matrix \(m\). All matrices in the question are assumed to be \(9 \times 9\) (indexes are in the range \([1..9]\)). All indices can be assumed to be of type \texttt{Int}, and the values should be of type \texttt{Integer}.

(a) [3 marks] Write a Haskell function \(\texttt{transpose}\) that takes a matrix and returns its transpose. The transpose of matrix \(m\) is a matrix \(m'\) such that \(m'_{ij} = m_{ji}\).

(b) [3 marks] Write a Haskell function \(\texttt{replace}\) that takes a matrix \(m\), indices \(i\) and \(j\) and a number \(v\), and returns a matrix that is the same as \(m\) but with the \((i,j)\)-th element having value \(v\).

(c) [4 marks] The inferred type of \(\texttt{transpose}\) in Haskell is not informative. What could be done in Haskell to make the type more informative? Show the actual Haskell code. Does your solution provide any extra type-checking in Haskell? Explain why or why not.