1. For each item, circle only the *best* answer. One correct answer=+3 points, one wrong answer=−1 point, 0 or mo than one answer=0 points. In the end, your score is mapped to an integer between 1 and 10.

   (a) In Haskell, we can change the value of a variable from within a function:

   A. always B. never C. sometimes D. none of the above

   (b) Given `f n = filter (\ x -> x < 2)`, the call `f 2 [10,20,30]` will yield:

   A. `[10,20,30]` B. `[]` C. `[0.2,0.1,6.666666666666667e-2]` D. `[5.0,10.0,15.0]`

   (c) The function defined by `fun f n = if (n=0) then 0 else f (n-1);` is:

   A. not tail recursive B. tail recursive C. not recursive D. recursive

   (d) Given $C = \lambda\, x\, y\, z\, .\, (x\ z)\ y$ $I = \lambda\, x\, .\, x$ $S = \lambda\, x\, y\, z\, .\, x\ z\ (y\ z)$, expression $I\ C\ S$ is equivalent to:

   A. $I\ (C\ S)$ B. $\lambda\, a.\ \lambda\, b.\ \lambda\, c.\ (b\ c)\ (a\ c)$ C. both of the above D. none of the above

2. Write the solution to this problem in the appropriate box. Each item is worth 3 points.

   (a) You are provided the following code:

```
u :: (a -> Bool) -> (a -> b) -> (a -> a) -> a -> [b]
u p h t x | p x       = []
          | otherwise = h x : u p h t (t x)

m :: (a->b) -> [a] -> [b]
m f xs = [ f x | x <- xs]
```

   Write appropriate lambda expressions for `p`, `h` and `t` such that, when passing them to `u`, this will behave similarly to `m f`.