This is a closed book exam. No calculators, cellphones, laptops, or other aids are permitted. Answer every question in the space that has been provided. You must show all your work without skipping steps; correct answers that are presented without justification may receive a mark of zero.

Student Name _____

Student Number _____

---

1. Perform a β-reduction on the following λ-Calculus expression. This process should require exactly 7 steps, and you will know you have finished when your expression is reduced to something of the form (λ■.(λ■.■)). (i.e., After reaching that point you will be unable to reduce it further).

[3.0 marks]

$$\lambda r.\left(\lambda s.\left(\lambda t.\left(\lambda u.\left(\lambda v.\left((t\ u)\ v\right)\right)\right)\ r\ r\ s\right)\left(\lambda w.\left(\lambda x.x\right)\right)\right)\left(\lambda y.\left(\lambda z.y\right)\right)$$

This is a closed book exam. No calculators, cellphones, laptops, or other aids are permitted. Answer every question in the space that has been provided. You must show all your work without skipping steps; correct answers that are presented without justification may receive a mark of zero.

Student Name _____

Student Number _____

2. In the space provided, define the term "lazy evaluation". [1.0 marks]

3. A "range" can be defined, using the three arguments "start", "stop", and "step", as the list of integers that represents the arithmetic sequence that begins at the "start" element and terminates without including the "stop" element using a common difference of "step". As a clarifying example, the range with "start" 3, "stop" 9, and "step" 2 would be represented with the list [3, 5, 7]. Write a recursive Haskell function (including a type declaration defined over Integers) that takes three arguments "start", "stop", and "step" (in that order) and computes the range using the definition provided. Your solution must be a single function (i.e., no helper functions) and may not call any other functions.
[4.0 marks]

This is a closed book exam. No calculators, cellphones, laptops, or other aids are permitted. Answer every question in the space that has been provided. You must show all your work without skipping steps; correct answers that are presented without justification may receive a mark of zero.

Student Name _____

Student Number _____

---

4. For this question you will implement (under specific constraints) a recursive function in Haskell that takes a list of characters as an argument and returns a list that contains only every third element from the argument list in the same order. As a clarifying example, this function, when passed the argument list "ABCDEFGHIJKLMNO", should return the list "CFILO". It should also be noted that your function must work (i.e., not terminate with an error) even if the number of elements in the argument list is less than three.

   You may NOT use the !! operator to complete this question, and you may NOT use any function that you haven't written here (i.e., you may NOT use any built-in functions, but you may use other functions if you write them yourself).

   a. Write this function (including a type declaration) by first writing a selector function (for retrieving the third element of a list) and then having your recursive function call your selector function as required. [4.0 marks]

   b. Write this function (including a type declaration) without selector functions (even if you wrote them yourself), using only pattern matching. [4.0 marks]

This is a closed book exam. No calculators, cellphones, laptops, or other aids are permitted. Answer every question in the space that has been provided. You must show all your work without skipping steps; correct answers that are presented without justification may receive a mark of zero.

Student Name _____

Student Number _____

5. For the purposes of this question, assume that the name "orderedPairs" refers to a list of all possible ordered pairs of integers, in RANDOM ORDER, where each component is between 0 and 5, inclusive. Using "orderedPairs" as a generator, write a list comprehension that produces a list of all possible ordered pairs of integers (in the order they appeared in the generator) for which the sum of the components is 6 and the first component is less than the second. You may use the fst and snd selector functions for this question if you wish, but your solution must be a list comprehension and cannot rely on any other functions. [2.0 marks]

6. Write a recursive function (including a type declaration) that takes a list of ordered pairs of integers as an argument and returns the same list you should receive from the list comprehension defined in 5 above. You may NOT use list comprehensions and you may NOT use any built-in functions, so your solution to this question may not call fst and snd unless you provide them yourself. [4.0 marks]

This is a closed book exam. No calculators, cellphones, laptops, or other aids are permitted. Answer every question in the space that has been provided. You must show all your work without skipping steps; correct answers that are presented without justification may receive a mark of zero.

Student Name _____

Student Number _____

---

7. For the purposes of this question, assume that the name "pixelData" refers to a list of 3-tuples of integers, where each 3-tuple (R, G, B) is a reference to the colour of a particular pixel in a raster image (as defined by the specification for your second assignment). Under the assumption that you knew, for certain, that the image being represented was SQUARE (i.e., had the same width and height), write Haskell functions (including a type declarations) that allow you repack this one-dimensional list of pixels into a SQUARE two-dimensional list of pixels. You may use the built-in length, take, and drop functions to complete this question if you wish, but you may NOT use any other built-in functions.

[3.0 marks]