0/10 Questions Answered

Practice Mid-Semester Exam 2

STUDENT NAME

Q1 Instructions
0 Points

Australian National University

You must acknowledge the following **integrity pledge** before proceeding. Check the boxes and enter your personal details.

- [ ] I am committed to being a person of integrity.
- [ ] I pledge, as a member of the Australian National University community, to abide by and uphold the standards of academic integrity outlined in the ANU statement on honesty and plagiarism, and I am aware of the relevant legislation, and understand the consequences of me breaching those rules.
- [ ] I will not actively communicate in any way with anyone else during this exam. This includes asking questions in any online forum.

Read and check off the following instructions:

1. You will need to be able to upload code files to this browser from your Haskell workspace. You must also **make sure all code you upload compiles without errors**.
   - [ ] Make sure this browser window is open on the machine where you plan to work.
   - [ ] If you are working on the ANU Linux VDI this browser window should be open in Linux, not on your personal machine.
   - [ ] If this browser window is not open where you plan to work close this window now and log back into the exam in a browser on your work machine.
2. This examination is timed.
   - [ ] Note the remaining time at the top right of this screen. Set an alarm for yourself if you need one.
3. Permitted materials. This is an open book exam. You might in particular find the course website, the Prelude documentation, and the Data.List documentation useful.
   - [ ] You may use any documentation you wish but **all work must be your own**.

Save Answer

Q2 Programming
2 Points

Von Neumann architecture allows

- Storing program code and data in separate memory types
- Storing program code and data in the same memory
- Storing program code in the primary memory but data in the secondary
- Storing data in the primary memory and program code in the secondary

Save Answer

Q3 Problem classes
2 Points

Given a matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

the inverse of the matrix exists, iff $ad - bc$ is not zero. Classify the problem of finding the existence of an inverse of A into one of the four classes of problems:

- Functional problems
- Decision problems
- Search problems
- Optimization problems

Save Answer

Q4 Sets
2 Points

If $B = \{\mathrm{F}, \mathrm{T}\}$ enumerate all elements of a set produced by $B + B + B$

- $\{\mathrm{F},\mathrm{F},\mathrm{F}\}, \{\mathrm{F},\mathrm{F},\mathrm{T}\}, \{\mathrm{F},\mathrm{T},\mathrm{F}\}, \{\mathrm{F},\mathrm{T},\mathrm{T}\}, \{\mathrm{T},\mathrm{F},\mathrm{F}\}, \{\mathrm{T},\mathrm{F},\mathrm{T}\}, \{\mathrm{T},\mathrm{T}$
- $\{1,\mathrm{F},\mathrm{F}\}, \{2,\mathrm{F},\mathrm{T}\}, \{3,\mathrm{T},\mathrm{F}\}, \{4,\mathrm{T},\mathrm{T}\}, \{5,\mathrm{F},\mathrm{F}\}, \{6,\mathrm{F},\mathrm{T}\}, \{7,\mathrm{T},\mathrm{F}$
- $\{1,\mathrm{F}\}, \{2,\mathrm{F}\}, \{3,\mathrm{F}\}, \{1,\mathrm{T}\}, \{2,\mathrm{T}\}, \{3,\mathrm{T}\}$
- $\{1,\mathrm{F},\mathrm{F}\}, \{1,\mathrm{F},\mathrm{T}\}, \{1,\mathrm{T},\mathrm{F}\}, \{1,\mathrm{T},\mathrm{T}\}, \{2,\mathrm{F},\mathrm{F}\}, \{2,\mathrm{F},\mathrm{T}\}, \{2,\mathrm{T},\mathrm{F}$

Save Answer

Q5 Basic types
2 Points

What is the type of the following value ([False, True], ['0', '1'])

- `([Bool], String)`
- `(Bool, Char)`
- `(Bool, [Char])`
- `([Bool], Char)`

Save Answer

Q6 Function currying
2 Points

The function `mod'` is defined as

```
mod' ::  Int -> Int -> Int
mod' x y = x `mod` y
```

Which of the following function calls is incorrect?

- result = (mod' 10)
- result = mod' 10 5
- result = mod' (10, 5)
- result = (mod' 10)(5)

Save Answer

Q7 Control structures
2 Points

A function `safetail :: [a] -> [a]` return a tail of a list and maps the empty list to itself.
Select all implementation that fits this function definition.

- [ ] `safetail xs = if null xs then [] else tail xs`
- [ ] `safetail xs | null xs = [] | otherwise = tail xs`
- [ ] `safetail xs = if null xs then tail xs else []`
- [ ] `safetail [] = []; safetail (_:xs) = xs`

Save Answer

Q8 List comprehensions
2 Points

The scalar product of two lists of integers $\mathtt{xs}$ and $\mathtt{ys}$ of length n is given by the sum of products of corresponding integers

$$\sum_{i=0}^{n-1} \mathtt{xs}_i * \mathtt{ys}_i$$

which of the following functions with type signature

```
scalarProduct :: [Double] -> [Double] -> Double
```

implements the scalar product:

- A.

```
scalarProduct l1 l2 = sum [x * y | (x, y) <- zip l1 l2]
```

- B.

```
scalarProduct   []  _   = 0
scalarProduct   _   []  = 0
scalarProduct (x:xs) (y:ys) = x*y + scalarProduct xs ys
```

- C.

```
scalarProduct l1 l2 = sum [x * y | x <- l1, y <- l2]
```

- D.

```
scalarProduct xs ys = if null xs || null ys
  then
      0
  else
      (head xs) * (head ys) + scalarProduct (tail xs) (tail ys)
```

- [ ] A
- [ ] B
- [ ] C
- [ ] D

Save Answer

Q9
2 Points

Upload a Haskell script that completes the following template (instructions included) for the function `sumdown`.

Cut and paste this template into a working file `Sumdown.hs` for you to edit and test using ghci before uploading:

```
module Sumdown where

-- | Returns the sum of the non-negative integers
-- from a given value down to zero.
--
-- No type signature for sumDown has been provided;
-- you must work this out for yourself.
--
-- Examples:
--
-- >>> sumdown 3
-- 6
sumdown = undefined // TODO
```

Please select file(s) | Select file(s)

Save Answer

Q10
2 Points

Upload a Haskell script that completes the following template (instructions included) for the function `selectElem`.

Cut and paste this template into a working file `SelectElem.hs` for you to edit and test using ghci before uploading:

```
module Selectelem where

-- | Return a list of elements of the input argument whose length
-- equal to the number of occurrences of the input argument in th
-- input list.
--
-- No type signature for selectElem has been provided;
-- you must work this out for yourself.
--
-- Examples:
--
-- >>> selectElem 3 [5,4,3,3,4,5,6]
-- [3,3]
-- >>> selectElem 'l' "Hello world!"
-- "lll"
selectElem = undefined // TODO
```

Please select file(s) | Select file(s)

Save Answer

Save All Answers | Submit & View Submission >