



IIT GOA

CS300 Programming Language Paradigms— Quiz 1
14 September 2022

Time: 1 hr
Total marks: 12
Total Pages: 2

Answer **all** questions. Make your answers short and precise.

1. (4 marks) What are the type of the following values? If the expression is invalid, briefly state why.

(a) `[1, '2', "3"]` `[Int, Char, String]` → Invalid

(b) `map not`

(c) `["x":[]]`

(d) `(1, 'x', [True])` `(Int, Char, [Bool])`

2. (2 marks) Give the type of the following functions.

(a) `pair (f, g) x = (f x, g x)`

(b) The function `takeWhile` when applied to a predicate p and a list xs returns the longest prefix (possibly empty) of xs consisting of elements that satisfy p . For example,

`»takeWhile (== True) [False, True]`
`[]` ✓

`»takeWhile (< 3) [1, 2, 3, 4, 1, 2, 3]`
`[1, 2]` ✓

3. (2 marks) Write a function `fm1` that takes a list as argument and returns a 3-tuple with the first, middle and last elements of the list. You may assume the list is non-empty. What is the type of `fm1`? Ord

4. (2 marks) Write a function `majority` which takes 3 boolean values and returns the majority among the three. For example,

`»majority True True False`
`True`

`»majority False True False`
`False`

majority

conditional

$\text{majority } x \ y \ z = \text{if } x = y \text{ then } x \text{ else } z$

Make sure your definition includes the type of majority

Boolean

Majority T T = T

T - T = T

- T T = T

- - - = F

5. (2 marks) Write a function `replaceHead` that takes a list and an element as argument and returns the list after replacing the head of the list with the second argument. For example,

» `replaceHead [1,2,3,4] 5`

`[5,2,3,4]`

» `replaceHead [] 2`

`[]`

Make sure your definition includes the type of `replaceHead`.

`replaceHead [1,2,3,4,5] 5`

`replaceHead xs n = n + 1`

TYPE :: `[a] -> Int -> [a]`

`replaceHead [] = []`

`replaceHead xs n = n + 1 temp`
where `temp = tail xs`

`re - exam`

`[a] -> a -> [a]`
a is argument here



IIT GOA

CS300 Programming Language Paradigms— Quiz 2

28 September 2022

Time: 1 hr

Total marks: 16

Total Pages: 2

Answer all questions. Make your answers short and precise.

1. (2+2+2+2 marks) Write the following functions using list comprehension.

(a) `noSpaces :: [Char] -> [Char]` that takes a string as argument and returns the string after eliminating all spaces.

```
>noSpaces "Hello World"
HelloWorld
```

(b) `numOfEs :: [Char] -> Int` that takes a string and returns the number of e's in the string.

```
>numOfEs "hello elephant"
3
```

(c) `nestedOdd :: [[Int]] -> [[Int]]` that takes a list of list of integers and returns it after removing all even integers. (Hint : Use nested list comprehension).

```
>nestedOdds [[1,2],[4,6],[6,7,8,9]]
[[1],[7,9]]
```

(d) `altMap :: (a -> b) -> (a -> b) -> [a] -> [b]` that takes two functions and a list and returns the list after applying the two functions alternatively to list elements.

```
>altMap (+1) (+2) [1..5]
[2,4,4,6,6]
```

2. (2+2+2+2 marks) Write the following functions using recursion.

(a) `kthElem :: [a] -> Int -> a` that takes a list and an integer k and returns the k th element of the list, where elements are numbered starting from 0. (Do not use the built-in operator !!).

```
>kthElem ['h','e','l','l','o'] 1
'e'
```

$[] - = \text{error}$
 $(x:xs) \cdot 0 = x$
 $kthElem (x:xs) n = kthElem xs (n-1)$

zip

k
o

(b) rotate :: [a] -> Int -> [a] that takes a list and an integer n and rotates the list n places to the left.

>rotate ['a','b', ['c','d','e']] 2
 ['c','d','e','a','b']

(c) dedup :: [a] -> [a] that eliminates consecutive duplicate elements in a list.

>dedup [2,4,4,4,6,6,8,4]
 [2,4,6,8,4]

(d) dropEvery :: [a] -> Int -> [a] that takes a list and an integer n and drops every nth element from the list.

>dropEvery [1,2,3,4,5,6,7,8,9,10] 3
 [1,2,4,5,7,8,10]



rotate :: [a] -> Int -> [a]

rotate [] = ~~error~~ []

rotate (x:xs) 0 = xs

rotate (x:xs) n = rotate (xs ++ [x]) (n-1)

dedup :: [a] -> [a]

dedup (x:x:xs) = dedup (x:xs)

dedup (x:s) = xs

dedup (x:xs) = x ++ dedup xs

dedup [] = []

dedup (xs) = dedup (xs)

dedup (x:y:xs) | x == y = dedup (y:xs)

2 of 2

| otherwise = x: dedup (y:xs)



I I T G O A

CS300 Programming Language Paradigms— Quiz 3

16 November 2022

Total marks: 15

Answer **all** questions.

1. (5 marks) Write a program that takes a string s and a number k as input and outputs a string where each letter in s is repeated k times. For example,

```
>./stutter
```

```
Enter a string:  hello
```

```
Enter a number:  3
```

```
Stuttered word:  hhheeeellllooo
```

2. (5 marks) Write a program that takes a file as command line argument and prints out only those lines which are not commented, i.e., lines that does not start with `//`. For example,

```
>cat foo.txt
```

```
//Introduction
```

```
Hi!  My name is xyz.
```

```
I am from abc.
```

```
//Education
```

```
I am a B.Tech graduate in computer science engineering.
```

```
>./removecomments foo.txt
```

```
Hi!  My name is xyz.
```

```
I am from abc.
```

```
I am a B.Tech graduate in computer science engineering.
```

3. (5 marks) Write an interactive program that takes an arithmetic expression in postfix notation separated by spaces and outputs the value of the expression.

```
>./calculator
```

```
2 3 * 5 4 - +
```

```
7
```

```
25 5 - 10 *
```

```
200
```



$$\underline{9.9} \quad \forall x (\neg p(x) \wedge q(x)) \vdash \forall x (p(x) \rightarrow q(x))$$

$$1. \forall x (\neg p(x) \wedge q(x)) \text{ (premise)}$$

$$2. \neg p(x_0) \wedge q(x_0) \quad (\forall x \text{ in } 1)$$

foldr (+) 0 - 3

[1,2,3]

→ - 3

- 3 - 2

f g x

2-3

3-0 43

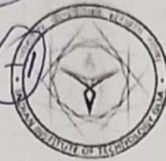
n: 4

g: a → b

f: a → b → c

2-3 =

1-1 = 0



IIT GOA

CS300 Programming Language Paradigms— Midsem

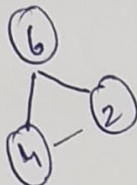
12 October 2022

Time: 2 hr

Total marks: 30

Total Pages: 3

Machine
algo de
Networks



Answer
by

Answer all questions. Make your answers short and precise.

b.

1. (2 marks) What are the types of the following values? If the expression is invalid, briefly state why.

(+) num = a → a
(a) (+1) \$ (0 <) (0 <) (num, ord a) = a → Bool
(b) subst f g x = f x (g x) (g) (a → b) → a → b

2. (2 marks) What are the values of these expressions?

foldl (-) 0 [1,2,3]

(a) foldl (-) 0 [1,2,3]

(b) foldr (-) 0 [1,2,3] 0 3-0

3. (2 marks) What is the most general type of the following function? Describe in brief what the function does with an example.

f [] ys = ys

f _ [] = []

f (x:xs) (y:ys) = f xs ys

taking 1

4. (3 marks) Write a function pairswap :: [a] → [a] that swaps the elements of a list pairwise, i.e., the first and second elements are swapped, the third and fourth are swapped and so on. Assume the list has an even number of elements and is possibly empty.

>pairswap [1,2,3,4,5,6]

[2,1,4,3,6,5]

5. (3 marks) A list of strings is connected if each string in the list (except the first) is obtained from the previous one by changing the character in exactly one position.

Define a function connected :: [String] → Bool that checks whether the input list of strings is connected.

[String] → Bool

1 of 3

→ Bool

ys

[] = []

true

false

(x:xs)

it

it

length xs

length = 1 = True

[pair(x,y)]

pair(x,y)

For example,

>connected []

True

>connected ['aa', 'ab', 'ba']

False

>connected ['aa', 'ab', 'bb', 'ba']

True

>connected ['ab', 'ab']

False

if length == 1 true
it

6. (3 marks) Write a function `combine :: [a] -> [[a]]` that groups consecutive duplicates into sublists. For example,

>combine [2,2,4,4,4,6]

[[2,2], [4,4,4], [6]]

7. (3 marks) Using `foldr`, write a function `separate :: [Char] -> ([Char], [Char])` that takes a string `s` and returns a 2-tuple with the digits and non-digits in the string `s` separated, with the initial order maintained. For example,

>separate "July 4, 1994"

("41994", "July , ")

Below is the partial implementation of `separate` using `foldr`. You need to write the folding function `f`.

`separate s = foldr f ([], []) s`

8. (4 marks) Write a function `scanl :: (a -> b -> a) -> a -> [b] -> [a]` that is the verbose version of the function `foldl`. It returns the list of successive values obtained by applying `foldl`. Some examples are given below.

>scanl (\ x y -> 2*x + y) 4 [1,2,3]

[4,9,20,43]

>scanl max 5 [1,3,4,6,7,8]

[5,5,5,5,6,7,8]

scanl

9. (4 marks) Using list comprehension, define a function `partitioned :: [Int] -> Bool` that returns True if there is an element `n` of the list such that:

- for each element `m` occurring before `n` in the list, $m \leq n$, and
- for each element `m` occurring after `n` in the list, $m > n$.

$$\underline{m \leq n}$$

For example,

>partitioned []

False

>partitioned [22]

True

>partitioned [19, 17, 18, 7]

False

>partitioned [7, 18, 17, 19]

True

>partitioned [19, 13, 16, 15, 19, 25, 22]

True

10. (4 marks) A *segment* of a list xs is a selection of adjacent elements of xs .

Define a function `segment :: [a] -> [[a]]` that takes a finite list xs as its argument and returns the list of all the segments of xs .

For example,

>segment []

[[]]

>segment [1, 2, 3]

[[1, 2, 3], [1, 2], [2, 3], [1], [2], [3]]

$$(x:xs) = []$$

[

segm

$$\text{segment } [] = []$$

$$\text{segment } (x:xs) =$$

$$\text{list } xs = \begin{bmatrix} [x, y], \dots \end{bmatrix} \mid x \leftarrow \text{head}(xs), y \leftarrow \text{tail}(xs)$$



IIT GOA

CS300 Programming Language Paradigms— Endsem - Part 1

30 November 2022

Time: 1 hour

Total marks: 10

Answer all questions. Make your answers short and precise.

$[a] \rightarrow a \rightarrow a \rightarrow a \rightarrow [a]$

1. (2 marks) What are the type of the following values? If the expression is invalid, briefly state why.

(a) `mys` where `mys = 2 : map (\x -> x + x) mys`

(b) `foo` where `foo g h = fmap (g.h)`

2. (2 marks) What is the most general type of the following function? Describe in brief (one or two sentence) what the function does.

`func xs = foldl (\v (x,y) -> if x==y then v else False) True ps`
 where `ps = zip xs (reverse xs)`

3. (2 marks) Define a function `removedups` using `foldr` that removes adjacent duplicates from a list. For example,

`>removedups [1,2,2,3,3,3,2,2,2,2]`
`[1,2,3,2]`

`removedups xs = foldr (f) [] xs`
`f x y = if x == y then y else x : y`

4. (2 marks) Given data declaration, data `Tree a = Leaf | Node [a] (Tree a) (Tree a)`. Determine the most general type for functions `i, j`.

`i Leaf = 'c'`
`i (Node x l r) = x`

`j Leaf = Leaf`
`j (Node x l r) = Node [length x] (j l) (j r)`

5. (2 marks) Define a new list datatype `Dualist a b` that can hold values of two different data types, in any order. For instance it would allow list such as `[1,2,'a',5,'b']` and `[False, 1,1, True]` that are otherwise not permitted by Haskell's built-in lists.

$[a] \rightarrow [Bool]$

`foo g h = fmap (g.h)`

- take a list
- Make list of 1st and last element in list

`map ::`

`(1 of 1)`

3

`foldl :: (a -> b -> a) -> a -> [b] -> [a]`



I I T G O A

CS300 Programming Language Paradigms— Endsem - Part 2

30 November 2022

Total marks: 15

Answer all questions.

1. (5 marks) Write a function `hammingDist :: Eq a => [a] -> [a] -> Maybe Int` that takes two list with equal length and returns the hamming distance between those list. Hamming distance between two list of equal length is the number of positions at which the corresponding elements are different. Use only prelude built-in functions or simple lambda expressions. For example,

```
>hammingDist [1,2,3,4] [1,3,4,4]
```

```
Just 2
```

```
>hammingDist "hello" "hello"
```

```
Just 0
```

```
>hammingDist "hello" "yellow"
```

```
Nothing
```

2. (5 marks) Write a program that takes a file as a command line argument and prints the contents after removing all the whitespaces. For example,

```
>cat foo.txt
```

```
Hi hello!
```

```
My name is xyz.
```

```
I am from abc.
```

```
>./removespace foo.txt
```

```
Hihello!Mynamaisxyz.Iamfromabc.
```

3. (5 marks) Write an interactive program that takes a string `s` consisting of letters `i` and `d` as input and outputs the difference in the number of `i`'s and `d`'s provided number of `i`'s \geq number of `d`'s for all prefixes of `s`. Your program should handle errors using Maybe monad. For example,

```
>./incDec
```

```
iiidid
```

```
Just 2
```

```
dddiid
```

```
Nothing
```

```
ididd
```

```
Nothing
```

```
iiiiidiadb
```

```
Nothing
```

