

Haskell Sheet

1. Find whether the following statements are True or False in Haskell, (Explain why?):

- a. `A = 3 :: Int`
- b. `mul (Int, Int) -> Int`
- c. `y :: double`
- d. `x = 3 :: Int`
`x = 7`
- e. `z :: [Int]`
`z = []`

2. Find one different way to rewrite the following statements in Haskell.

- a. `sum :: Int -> Int -> Int.`
- b. `y :: Int`
`y = 3`
- c. `greaterThan100 :: [Integer] -> [Integer]`
`greaterThan100 xs = filter (>100) xs`
- d. `funF :: String -> Int`
`funF str = case str of`
 `[] -> 3`
 `('H':s) -> length s`
 `_ -> 7`
- e. `funG :: [Integer] -> Bool`
`funG = even . length`

3. Use the following Haskell program to answer the questions

```
data FailableDouble = Failure
    | Ok Double
    deriving Show

safeDiv :: Double -> Double -> FailableDouble
safeDiv _ 0 = Failure
safeDiv x y = Ok (x / y)
```

- What does Haskell return for the query: `*main> safeDiv 4 2`.
- What does Haskell return for the query: `*main> safeDiv 7 0`.
- Define 2 lists of type `FailableDouble` and initialize each with three values.
- What does Haskell return if we pass the two lists in (c) as parameters to `safeDiv`.

4. Use the following Haskell program to answer the questions: .

```
1 anonymous :: [Int] -> Int
2 anonymous [] = 0
3 anonymous (h:t) = 1 + anonymous t
```

- What does Haskell return for the query: `*main> anonymous [2, 7, 1, 6, 12]` .
- What does Haskell return for the query: `*main> anonymous 7`.
- If we removed line (1), does the program still work?
- If we removed line (2), What does Haskell return for the same query in (a).
- Suggest a better name instead of “anonymous”.

5. Given user defined data type “IntList” and two recursive function “addOneToAll”, and “squareAll” as following:

```
data IntList = Empty
              | Cons Int IntList
  deriving Show

addOneToAll :: IntList -> IntList
addOneToAll Empty = Empty
addOneToAll (Cons x xs) = Cons (x + 1) (addOneToAll xs)

squareAll :: IntList -> IntList
squareAll Empty = Empty
squareAll (Cons x xs) = Cons (x * x) (squareAll xs)
```

- What does Haskell return if we apply addOneToAll and squareAll to “myIntList”, where
myIntList = Cons 2 (Cons (-3) (Cons 5 Empty))
- Write a recursion pattern to replace the two given functions.
- In the given code “IntList” includes only integer values. Rewrite IntList to accept different data types such as Int, Double, Float, ... etc.

6. Given the following Haskell program:

```
f :: [Integer] -> Integer
f [] = 0
f (x:xs)
  | x > 3 = (7*x + 2) + f xs
  | otherwise = f xs
```

- What does Haskell return for the query: *main> f [1, 5, -1, 4].
- Rewrite the previous program in a wholemeal (point-free) style.

7. Write the Haskell function “fact” to calculate the Factorial of a given integer number n. (Hint: Factorial n , $n! = n * (n-1) * (n-2) * \dots * 1$).

Sample run:

```
*main> fact 4
24
```

8. Write the Haskell function “oddList” to accept a list of Integers as an input and return another list with only Odd numbers from the input list.

Sample run:

```
*main> oddList [1,2,5,8,3,4]
[1,5,3]
```

Do your best; Good Luck.