

Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

1. Write a script to **find the last but one element** of a list. [Input: [1,2,3,4] Output: 3]

```
1.hs U X
PPL > Week-4 > 1.hs > ...
1  --Write a script to find the last but one element of a list. [Input: [1,2,3,4] Output: 3]
2
3  lastButOne :: [a] -> a
4  lastButOne [] = error "Empty list"
5  lastButOne [x] = error "List has only one element"
6  lastButOne [x,_] = x
7
8  lastButOne (_,xs) = lastButOne xs
9
10 -- AM.EN.U4CSE20349
11 -- main function
12 main :: IO ()
13 main = do
14     print(lastButOne [1,2,3,4])

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS
(base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/1.hs"
3
(base) aj@RAJs-MacBook-Air S6 %
```

2. Write a script to **find the kth element of list**, where k is the index. [Input: [1,2,3,4] k=2 Output: 3]

```
1.hs U 2.hs U X
PPL > Week-4 > 2.hs > main
1  import GHC.Exts.Heap (GenClosure(key))
2  -- Write a script to find the kth element of list, where k is the index. [Input: [1,2,3,4] k=2 Output: 3]
3
4  kthElement :: [a] -> Int -> a
5  kthElement [] _ = error "Empty list"
6  kthElement (x:xs) 1 = x
7  kthElement (x:xs) k = kthElement xs (k-1)
8
9  -- AM.EN.U4CSE20349
10 -- main function
11 main :: IO ()
12 main = do
13     print(kthElement [1,2,3,4] 2)

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS
(base) aj@RAJs-MacBook-Air S6 %
```

Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

3. Write a script to find out whether a list is a **palindrome**. [Input1: [1,2,1] Output: True, Input2="mom" Output2:True]

```
1 --Write a script to find out whether a list is a palindrome. [Input1: [1,2,1] Output: True, Input2="mom" Output2:True]
2
3 isplindrome :: [a] -> Bool
4 isplindrome [] = True
5 isplindrome [x] = True
6 isplindrome (x:xs) = if x == last xs then isplindrome (init xs) else False
7
8 -- AM.EN.U4CSE20349
9 -- main function
10 main = do
11   print(isplindrome [1,2,1])
12   print(isplindrome "mom")
```

4. Write a script to **remove duplicates** from a given list. [Input : [1,1,2,3] Output: [1,2,3]]

```
1 --Write a script to remove duplicates from a given list. [Input : [1,1,2,3] Output: [1,2,3]]
2
3 removeDuplicates :: (Eq a) => [a] -> [a]
4 removeDuplicates [] = []
5 removeDuplicates (x:xs) = x : removeDuplicates (filter (/= x) xs)
6
7 -- AM.EN.U4CSE20349
8 -- main function
9 main = do
10   print(removeDuplicates [1,1,2,3])
11
```

```
• (base) aj@RAJs-MacBook-Air S6 % runhsc "/Users/aj/Desktop/S6/PPL/Week-4/4.hs"
[1,2,3]
(base) aj@RAJs-MacBook-Air S6 %
```

Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

5. Define a function duplicate which will **duplicate each element of the list** and produce a new list. Example [1,2,3] - will give output [1,1,2,2,3,3].

```
1 -- Define a function duplicate which will duplicate each element of the list and produce a new list. Example [1,2,3]
2
3 duplicate :: [a] -> [a]
4 duplicate [] = []
5 duplicate (x:xs) = x:x:duplicate xs
6
7 -- AM.EN.U4CSE20349
8 -- main function
9 main = do
10     print(duplicate [1,2,3])
11
```

```
(base) aj@RAJs-MacBook-Air S6 % runhsc "/Users/aj/Desktop/S6/PPL/Week-4/5.hs"
[1,1,2,2,3,3]
(base) aj@RAJs-MacBook-Air S6 %
```

6. Define a function to **replicate the elements of a list n times**. Let [1,2] be a list and let n be 3, then the resultant list will be [1,1,1,2,2,2].

```
1 -- Define a function to replicate the elements of a list n times.
2 -- Let [1,2] be a list and let n be 3, then the resultant list will be [1,1,1,2,2,2].
3
4 replicated :: [a] -> Int -> [a]
5 replicated [] _ = []
6 replicated (x:xs) n = replicateHelper x n ++ replicated xs n
7
8 replicateHelper :: a -> Int -> [a]
9 replicateHelper _ 0 = []
10 replicateHelper x n = x:replicateHelper x (n-1)
11
12 -- AM.EN.U4CSE20349
13 -- main function
14 main = do
15     print(replicated [1,2,3] 3)
```

```
(base) aj@RAJs-MacBook-Air S6 % runhsc "/Users/aj/Desktop/S6/PPL/Week-4/6.hs"
[1,1,1,2,2,2,3,3,3]
(base) aj@RAJs-MacBook-Air S6 %
```

Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

7. Write a function to **remove every nth element** from the list. Let **[1,2,3,4,5,6,7,8,9,10]** be the list and value of **n** be **3**, then the resultant list will be **[1,2,4,5,7,8,10]**(after every nth element of the input list is **[1,2,3,4,5,6,7,8,9,10]** removed).

```
1 -- Write a function to remove every nth element from the list.
2
3 removeNth :: [a] -> Int -> [a]
4 removeNth [] _ = []
5 removeNth xs 1 = []
6 removeNth (x:xs) n = x : removeNth (drop (n-1) xs) n
7
8 -- main function
9 main :: IO ()
10 main = do
11   print(removeNth [1,2,3,4,5,6,7,8,9,10] 3)
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```
• (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/7.hs"
[1,4,7,10]
• (base) aj@RAJs-MacBook-Air S6 %
```

8. **Split a list** by defining a function which takes an input **list** and an **integer n** and divides the list into two the **first n elements as the first list** and **the rest as the second list** and form a list of lists. Let **[1..10]** be a list and value of **n** be **4** then the new list formed is **[1,2,3,4],[5,6,7,8,9,10]**. Another example I/P splits "amr" 4 & O/P- **["amr", ""]**

```
1 -- Split a list by defining a function which takes an input list
2 -- and an integer n and divides the list into two parts.
3
4 splitlist :: [a] -> Int -> ([a],[a])
5 splitlist [] _ = ([],[])
6 splitlist xs 0 = ([],xs)
7 splitlist (x:xs) n = (x:ys,zs)
8   where (ys,zs) = splitlist xs (n-1)
9
10 -- main function
11 main :: IO ()
12 main = do
13   print(splitlist [1,2,3,4,5,6,7,8,9,10] 3)
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```
• (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/8.hs"
([1,2,3],[4,5,6,7,8,9,10])
• (base) aj@RAJs-MacBook-Air S6 %
```

Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

9. Define a function that will **slice a list** based on the input indices **i** and **k**. Consider a list **[1..10]** and let **i = 2** and **k = 4** respectively then the resultant list will be **[3,4,5]**.

```
1 -- Define a function that will slice a list based on the input indices i and k.
2
3 slice :: [a] -> Int -> Int -> [a]
4 slice [] _ _ = []
5 slice (x:xs) 1 1 = [x]
6 slice (x:xs) 1 k = x:slice xs 1 (k-1)
7 slice (x:xs) i k = slice xs (i-1) (k-1)
8
9 -- main function
10 main = do
11     print(slice [1,2,3,4,5,6,7,8,9,10] 3 7)
12
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```
• (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/9.HS"
[3,4,5,6,7]
• (base) aj@RAJs-MacBook-Air S6 %
```

10. Define a function to **remove the kth indexed element** from a list. Consider a list **[1..10]**, and value of **n** be **2**, then resultant list will be **[1,2,4,5,6,7,8,9,10]**.

```
1 -- Define a function to remove the kth indexed element from a list.
2
3 removeKth :: [a] -> Int -> [a]
4 removeKth [] _ = []
5 removeKth (x:xs) 1 = xs
6 removeKth (x:xs) k = x:removeKth xs (k-1)
7
8 -- main function
9 main = do
10     print(removeKth [1,2,3,4,5,6,7,8,9,10] 3)
11
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```
• (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/10.HS"
[1,2,4,5,6,7,8,9,10]
• (base) aj@RAJs-MacBook-Air S6 %
```

Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

11. Define a function that will **insert an element n at a particular index i** of a list xs. Let xs=[1..10], i=2, n=11, then the output will be [1,2,11,3,4,5,6,7,8,9,10].

```
1 -- Define a function that will insert an element n at a particular index i of a list xs.
2
3 insertAt :: [a] -> a -> Int -> [a]
4 insertAt [] _ _ = []
5 insertAt (x:xs) n 1 = n:x:xs
6 insertAt (x:xs) n i = x:insertAt xs n (i-1)
7
8 -- main function
9 main = do
10   print(insertAt [1,2,3,4,5,6,7,8,9,10] 11 3)
```

```
(base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/11.HS"
[1,2,11,3,4,5,6,7,8,9,10]
(base) aj@RAJs-MacBook-Air S6 %
```

1. Define a function that takes an integer number **n** and returns the list of the **first n prime numbers**.

```
1 -- Define a function that takes an integer number n and returns the list of the first n prime numbers.
2
3 prime :: Int -> [Int]
4 prime 0 = []
5 prime n = primeHelper 2 n
6
7 primeHelper :: Int -> Int -> [Int]
8 primeHelper _ 0 = []
9 primeHelper x n = if isPrime x then x:primeHelper (x+1) (n-1) else primeHelper (x+1) n
10
11 isPrime :: Int -> Bool
12 isPrime 1 = False
13 isPrime 2 = True
14 isPrime x = if length [y | y <- [2..x-1], x `mod` y == 0] == 0 then True else False
15
16 -- AM.EN.U4CSE20349
17 -- main function
18 main = do
19   print(prime 10)
```

```
(base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/12.hs"
[2,3,5,7,11,13,17,19,23,29]
(base) aj@RAJs-MacBook-Air S6 %
```


Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

2. Define a predicate to verifies whether a list is sorted in **ascending order**.

```
1  -- Define a predicate to verifies whether a list is sorted in ascending order.
2
3  isSorted :: (Ord a) => [a] -> Bool
4  isSorted [] = True
5  isSorted [x] = True
6  isSorted (x:y:xs) = if x <= y then isSorted (y:xs) else False
7
8  -- AM.EN.U4CSE20349
9  -- main function
10 main :: IO ()
11 main = do
12     print(isSorted [1,2,3])
13     print(isSorted [1,3,2])
14     print(isSorted [1,2,3,4,5,6,7,8,9,10])
15     print(isSorted [1,2,3,4,5,6,7,8,9,10,1])
16     print(isSorted [1,2,3,4,5,6,7,8,9,10,11])
17     print(isSorted [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15])
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```
• (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/13.hs"
True
False
True
False
True
True
• (base) aj@RAJs-MacBook-Air S6 %
```

3. Define a function **myCons** that behaves as `:` but is defined in terms of `++`.

```
1  -- Define a function myCons that behaves as : but is defined in terms of ++.
2
3  myCons :: a -> [a] -> [a]
4  myCons x xs = [x] ++ xs
5
6  -- AM.EN.U4CSE20349
7  -- main function
8  main :: IO ()
9  main = do
10     print(myCons 1 [2,3,4,5])
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```
• (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/14.hs"
[1,2,3,4,5]
• (base) aj@RAJs-MacBook-Air S6 %
```

Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

4. Define a function that takes an integer **number n** and a **value v** and creates a list containing **n occurrences of v**.

```
2.hs U 3.hs 1, U 4.hs U 5.hs U 6.hs U 8.hs U 1.1.hs U 12.hs U 13.hs U 14.hs U 15.hs U
PPL > Week-4 > 15.hs > main
1 -- Define a function that takes an integer number n and a value v and creates a list containing n occurrences of v
2
3 replicate' :: Int -> a -> [a]
4 replicate' 0 _ = []
5 replicate' n x = x:replicate' (n-1) x
6
7
8 -- main function
9 main :: IO ()
10 main = do
11     print(replicate' 10 5)
12
13
14 PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS
* (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/15.hs"
[5,5,5,5,5,5,5,5,5,5]
(base) aj@RAJs-MacBook-Air S6 %
```

5. To find all the digits in a string where the prelude function

```
2.hs U 3.hs 1, U 4.hs U 5.hs U 6.hs U 8.hs U 1.1.hs U 12.hs U 13.hs U 14.hs U 6-1.hs U
PPL > Week-4 > 6-1.hs > ...
1 -- To find all the digits in a string
2
3 import Data.Char (isDigit)
4 import Data.Char
5 digits :: String -> String
6 digits [] = []
7 digits (x:xs) = if isDigit x then x : digits xs else digits xs
8
9 -- main function
10 main :: IO ()
11 main = do
12     print(digits "Hello 123 World 456")
13     print(digits "Hello World")
14
15
16 PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS
* (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/6-1.hs"
"123456"
""
(base) aj@RAJs-MacBook-Air S6 %
```


Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

6. Give a definition of a function which **doubles all** the elements of a list of integers

```
2.hs U 3.hs 1,U 4.hs U 5.hs U 6.hs U 8.hs U 1.1.hs U 12.hs U 13.hs U 14.hs U 7-1.hs U
PPL > Week-4 > 7-1.hs > ...
1 -- Give a definition of a function which doubles all the elements of a list of integers
2
3 doubleAll :: [Int] -> [Int]
4 doubleAll [] = []
5 doubleAll (x:xs) = (x*2):doubleAll xs
6
7 -- AM.EN.U4CSE20349
8 -- main function
9 main :: IO ()
10 main = do
    print(doubleAll [1,2,3])

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS
(base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/7-1.hs"
[2,4,6]
(base) aj@RAJs-MacBook-Air S6 %
```

7. Give a definition of a function which **converts all small letters** in a String **into capitals**, leaving the other characters unchanged.

```
3.hs 1,U 4.hs U 5.hs U 6.hs U 8.hs U 1.1.hs U 12.hs U 13.hs U 14.hs U 7-1.hs U 8-1.hs U
PPL > Week-4 > 8-1.hs > toUpperAll
1 -- Give a definition of a function which converts all small letters in a String into capitals,
2 -- leaving the other characters unchanged.
3
4 import Data.Char (isLower, toUpper)
5
6 toUpperAll :: String -> String
7 toUpperAll [] = []
8 toUpperAll (x:xs) = if isLower x then toUpper x : toUpperAll xs else x : toUpperAll xs
9
10 -- AM.EN.U4CSE20349
11 -- main function
12 main :: IO ()
13 main = do
14     print(toUpperAll "Hello 123 World 456")
15     print(toUpperAll "Hello World")

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS
(base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/8-1.hs"
"HELLO 123 WORLD 456"
"HELLO WORLD"
(base) aj@RAJs-MacBook-Air S6 %
```

Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

8. How would you modify above function to give which behaves in the same way except that all non-letters are removed from the list? You should check the Char.hs library to see whether it contains any functions useful in solving this problem.

```
4.hs U 5.hs U 6.hs U 8.hs U 1.1.hs U 12.hs U 13.hs U 14.hs U 7-1.hs U 8-1.hs U 9-1.hs U
PPL > Week-4 > 9-1.hs > main
1 -- all non-letters are removed from the list
2
3 import Data.Char (isLetter)
4 import Data.Char
5
6 removeNonLetters :: String -> String
7 removeNonLetters [] = []
8 removeNonLetters (x:xs) = if isLetter x then x : removeNonLetters xs else removeNonLetters xs
9
10 -- main function
11 main :: IO ()
12 main = do
13     print(removeNonLetters "Hello 123 World 456")
14     print(removeNonLetters "Hello World")
15
16 PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS
• (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/9-1.hs"
"HelloWorld"
"HelloWorld"
• (base) aj@RAJs-MacBook-Air S6 %
```

9. Define the function which **returns** the **list of divisors** of a **positive integer** (and the empty list for other inputs).

```
4.hs U 5.hs U 6.hs U 8.hs U 1.1.hs U 12.hs U 13.hs U 14.hs U 7-1.hs U 8-1.hs U 10-1.hs U
PPL > Week-4 > 10-1.hs > ...
1 -- which returns the list of divisors of a positive integer (and the empty list for other inputs).
2
3 divisors :: Int -> [Int]
4 divisors 0 = []
5 divisors x = [y | y <- [1..x], x `mod` y == 0]
6
7 -- main function
8 main :: IO ()
9 main = do
10     print(divisors 10)
11
12 PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS
• (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/10-1.hs"
[1,2,5,10]
• (base) aj@RAJs-MacBook-Air S6 %
```

Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

10. Define the function which picks out all occurrences of an integer n in a list. For instance,

```
4.hs U 5.hs U 6.hs U 8.hs U 11.hs U 12.hs U 13.hs U 14.hs U 7-1.hs U 8-1.hs U 11-1.hs U
PPL > Week-4 > 11-1.hs > ...
1 -- which picks out all occurrences of an integer n in a list of integers
2
3 occurrences :: Int -> [Int] -> [Int]
4 occurrences _ [] = []
5 occurrences n (x:xs) = if n == x then x:occurrences n xs else occurrences n xs
6
7 -- AM.EN.U4CSE20349
8 -- main function
9 main :: IO ()
10 main = do
11     print(occurrences 2 [1,2,3,2,2,2,3,4,5] )
12
13 PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS
• (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/11-1.hs"
[2,2,2,2]
◦ (base) aj@RAJs-MacBook-Air S6 %
```

11. Using matches or otherwise, define a function which is **True** if the **Int** is an **element** of the list, and **False** otherwise.

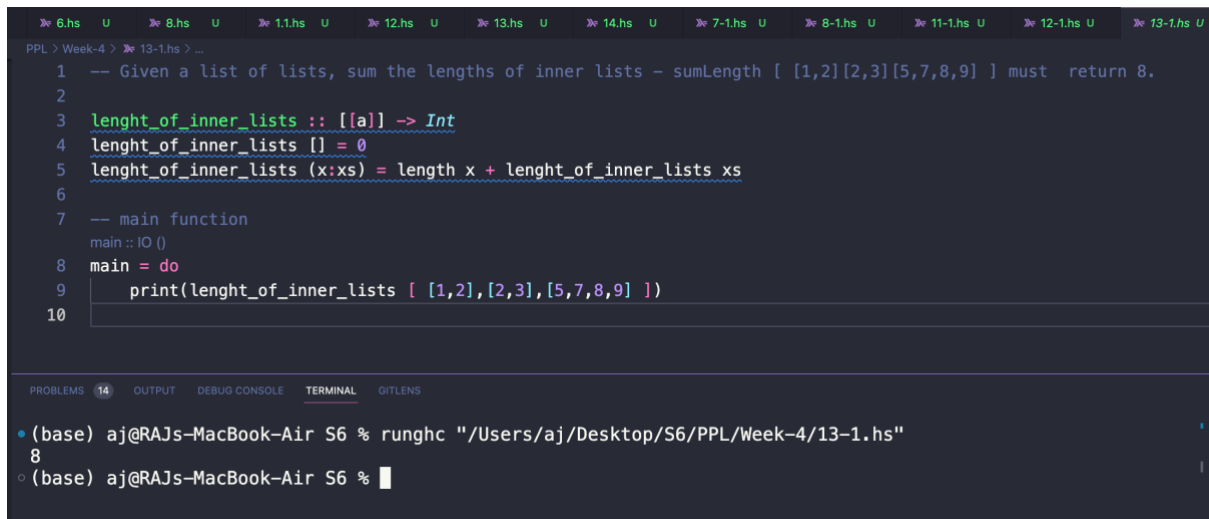
```
5.hs U 6.hs U 8.hs U 11.hs U 12.hs U 13.hs U 14.hs U 7-1.hs U 8-1.hs U 11-1.hs U 12-1.hs U
PPL > Week-4 > 12-1.hs > ...
1 -- which is True if the Int is an element of the list, and False otherwise.
2
3 isElem :: Int -> [Int] -> Bool
4 isElem _ [] = False
5 isElem x (y:ys) = if x == y then True else isElem x ys
6
7 -- AM.EN.U4CSE20349
8 -- main function
9 main :: IO ()
10 main = do
11     print(isElem 1 [1,2,3])
12     print(isElem 4 [1,2,3])
13     print(isElem 1 [1,2,3,4,5,6,7,8,9,10])
14     print(isElem 4 [1,2,3,5,6,7,8,9,10,11])
15     print(isElem 1 [1,2,3,4,5,6,7,8,9,10,11])
16     print(isElem 1 [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15])
17
18 PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS
• (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/12-1.hs"
True
False
True
False
True
True
True
◦ (base) aj@RAJs-MacBook-Air S6 %
```

Lab Sheet – 5

PATEL RAJKUMAR PANKAJBHAI

AM.EN.U4CSE20349

12. Given a list of lists, sum the lengths of inner lists - `sumLength [[1,2][2,3][5,7,8,9]]` must return 8



```

1  -- Given a list of lists, sum the lengths of inner lists - sumLength [ [1,2][2,3][5,7,8,9] ] must return 8.
2
3  lenght_of_inner_lists :: [[a]] -> Int
4  lenght_of_inner_lists [] = 0
5  lenght_of_inner_lists (x:xs) = length x + lenght_of_inner_lists xs
6
7  -- main function
8  main :: IO ()
9  main = do
10     print(lenght_of_inner_lists [ [1,2],[2,3],[5,7,8,9] ])

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```

• (base) aj@RAJs-MacBook-Air S6 % runghc "/Users/aj/Desktop/S6/PPL/Week-4/13-1.hs"
8
◦ (base) aj@RAJs-MacBook-Air S6 %

```