LABSHEET – 8

SHOAIB AKHTAR
AM.EN.U4CSE20163

1. Use **foldr** and **foldl** to define functions **lengthr** and **lengthl** respectively to find the number of elements in a list.

```
ghci> :l lab.hs
[1 of 1] Compiling Main             ( lab.hs, interpreted )
Ok, one module loaded.
ghci> lengthr [1,2,3,4,5]
5
ghci> lengthl [4,58,7,8,9,55,7,8,5]
9
ghci>
```

2. Find the smallest element of a list by defining functions **minr** and **minl** with the implementation of **foldr** and **foldl** respectively.

```
ghci> minr [1,2,3,4,5]
1
ghci> minl [4,58,7,8,9,55,7,8,5]
4
ghci>
```

3. Using **foldr**, define a function to reverse the current list.

```
ghci> reverse' [4,8,9,2,4,33,66,1,2,7,5]
[5,7,2,1,66,33,4,2,9,8,4]
ghci>
```

4. Define a function **remover** using **foldr** which takes two strings as its arguments and removes every letter from the second list that occurs in the first list. For example, remove "ece" "cse" = "s".

```
ghci> remover "ece" "cse"
"s"
ghci>
```

5. Define a function **rmdups** which removes adjacent duplicates from a list. For example, rmdups [1,2,2,3,3,3,1,1] = [1,2,3,1]. Then implement new versions of rmdups using **foldr** and **foldl** respectively. [Total 3 functions to be defined.]

```
ghci> rmdups [1,2,2,3,3,3,1,1]
[1,2,3,1]
ghci> rmdupsr [1,2,2,3,3,3,1,1]
[1,2,3,1]
ghci> rmdupsl [1,2,2,3,3,3,1,1]
[1,3,2,1]
ghci>
```

6. Define a function **approxe n** using foldl such that

```
ghci> approxe 7
2.7182539682539684
ghci>
```

7. Define the following function mult using lambda expressions. Show the type of lambda expression.

```
ghci> :l lab.hs
[1 of 1] Compiling Main
Ok, one module loaded.
ghci> mult 4 5 2
40
ghci>
```

8. Define the following add function using lambda expressions:

```
[1 of 1] Compiling Main
Ok, one module loaded.
ghci> add (2,3) (4,5)
(6,8)
ghci>
```

9. Using Lamda expression check whether an input list is palindrome or not.

```
ghci> isPalindrome [1,2,3,3,2,1]
True
ghci>
```

10. Use the defined function of Q9 to apply on a list of lists - where each list in the list will be checked if palindrome or not.

```
ghci> checkPalindromes [[1],[2,3],[2,3],[1]]
[True,False,False,True]
ghci>
```