**Федеральное агентство связи**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования «Московского технического университета связей и информатики»**

Кафедра Математической кибернетики и информационных технологий

Лабораторная работа №4

Работа с файлами в Haskell

Выполнил:

Студент группы БВТ1702

Рыбаков Д. В.

Вариант 19

1. **Формулировка задания:** разработать программу, которая выполняет данную задачу: программа считывает две матрицы из файлов и записывает в третий файл матрицу, являющуюся их суммой.

**ТЗ:**

1. Считать матрицы из двух входных файлов.
2. Поэлементно сложить эти матрицы
3. Вывести сложенные матрицы в файл
4. **Код программы**

import System.IO

import System.Directory

import Data.List

import Data.Char

import Data.List.Split

main = do

handle1 <- openFile "matrix1.txt" ReadMode

contents1 <- hGetContents handle1

let matrix1 = lines contents1

matr1 = matrix1

handle2 <- openFile "matrix2.txt" ReadMode

contents2 <- hGetContents handle2

let matrix2 = lines contents2

matr2 = matrix2

outHandle <- openFile "out.txt" WriteMode

putStrLn "These are your first matrix:"

putStr $ unlines matr1

putStrLn "Here is your second matrix:"

putStr $ unlines matr2

let dimentions1 = splitOn " " (matr1 !! 0)

let dimentions2 = splitOn " " (matr2 !! 0)

let rows1 = digitToInt $ head $ dimentions1!!0

let cols1 = digitToInt $ head $ dimentions1!!1

let rows2 = digitToInt $ head $ dimentions2!!0

let cols2 = digitToInt $ head $ dimentions2!!1

if rows1 == rows2 && cols1 == cols2

then do

-- let s = customZipWith (splitOn " " (matr1 !! 1)) (splitOn " " (matr2 !! 1))

-- putStrLn $ matrixOut s

iterator outHandle matr1 matr2 rows1 1

else do

hPutStrLn outHandle "These matrices are not equat thus cannot be summed"

hClose handle1

hClose handle2

hClose outHandle

return ()

matrixOut :: [Int] -> String

matrixOut [] = ""

matrixOut (x:xs) = show x ++ " " ++ matrixOut xs

iterator outHandle matr1 matr2 rows count = do

let s = customZipWith (splitOn " " (matr1 !! count)) (splitOn " " (matr2 !! count))

hPutStrLn outHandle $ matrixOut s

if count == rows then do return ()

else do iterator outHandle matr1 matr2 rows (count+1)

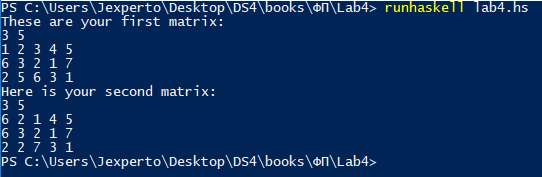
customZipWith :: [[Char]] -> [[Char]] -> [Int]

customZipWith [] \_ = []

customZipWith \_ [] = []

customZipWith (x:xs) (y:ys) = ((digitToInt $ head x) + (digitToInt $ head y)) : customZipWith xs ys

1. **Контрольное тестирование**



Файл matrix1.txt:

3 5

1 2 3 4 5

6 3 2 1 7

2 5 6 3 1

Файл matrix2.txt:

3 5

6 2 1 4 5

6 3 2 1 7

2 2 7 3 1

Файл out.txt:

7 4 4 8 10

12 6 4 2 14

4 7 13 6 2