**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

**Лабораторна робота № 3**

з дисципліни «Непроцедурне Програмування »

Тема роботи: «Функції для роботи з рядками»

Виконав студент

групи AнД-31

Швед І.Г.

Перевірив:

Гайна Г.А.

**Київ – 2023**

**Записати функцію, яка реалізує дану команду, навести приклад.**

1. Взяття N-го елементу з даного списку.

myList1 = [1..5]

element = myList1 !! 4

1. Функція додавання елементів двох списків

addLists :: [a] -> [a] -> [a]

addLists list1 list2 = list1 ++ list2

addedList = addLists myList1 myList2

1. Функція перестановки місцями парних та непарних елементів даного списку.

swapEvenOdd :: [a] -> [a]

swapEvenOdd [x] = [x]

swapEvenOdd (x:y:rest) = y : x : swapEvenOdd rest

swapList = swapEvenOdd myList1

1. Обертання списку.

reversedList = reverse myList1

1. Функція, яка показує перше входження даного елементу в список.

findFirstIndex :: (Eq a) => a -> [a] -> Maybe Int

findFirstIndex element list = elemIndex element list

elementToFind = 4

findedElement = findFirstIndex elementToFind myList1

1. Функція включення даного атому в дану позицію.

insertAtIndex :: Int -> a -> [a] -> [a]

insertAtIndex n element list = take n list ++ [element] ++ drop n list

insertPosition = 2

elementToInsert = 14

insertedList = insertAtIndex insertPosition elementToInsert myList2

1. Функція, яка визначає для кожного атома частоту його входження.

countFrequency :: (Eq a) => [a] -> [(a, Int)]

countFrequency list = [(x, length (filter (== x) list)) | x <- nub list]

frequencyList = countFrequency myList3

1. Функція, що переводить малі літери в Великі «абв»- «АБВ»

convertToUpperCase :: String -> String

convertToUpperCase str = map toUpper str

inputString = "abcDf"

upperString = convertToUpperCase inputString

1. Функція, що видаляє перше входження даного елементу до списку.

removeFirstOccurrence :: (Eq a) => a -> [a] -> [a]

removeFirstOccurrence element list = delete element list

elementToRemove = 3

removedList = removeFirstOccurrence elementToRemove myList3

1. Функція, що перетворює список пар даного списку окремо в список перших і других компонентів.

splitPairs :: [(a, b)] -> ([a], [b])

splitPairs pairs = unzip pairs

myPairs = [(1, 'a'), (2, 'b'), (3, 'c')]

(firstComponents, secondComponents) = splitPairs myPairs

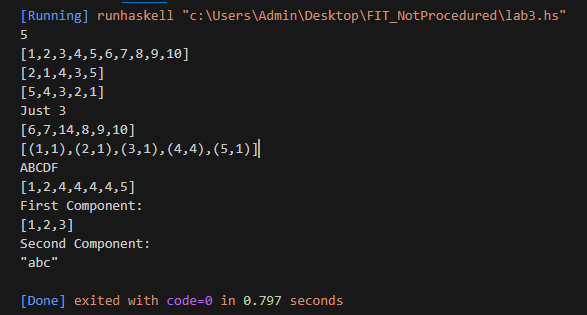
**Усі списки, використані в програмі:**

myList1 = [1..5]

myList2 = [6..10]

myList3 = [1,2,3,4,4,4,4,5]

**Результат роботи всіх програм:**



**Код роботи:**

import Data.List (elemIndex)

import Data.List (nub)

import Data.Char (toUpper)

import Data.List (delete)

myList1 = [1..5]

myList2 = [6..10]

myList3 = [1,2,3,4,4,4,4,5]

element = myList1 !! 4

addLists :: [a] -> [a] -> [a]

addLists list1 list2 = list1 ++ list2

addedList = addLists myList1 myList2

swapEvenOdd :: [a] -> [a]

swapEvenOdd [x] = [x]

swapEvenOdd (x:y:rest) = y : x : swapEvenOdd rest

swapList = swapEvenOdd myList1

reversedList = reverse myList1

findFirstIndex :: (Eq a) => a -> [a] -> Maybe Int

findFirstIndex element list = elemIndex element list

elementToFind = 4

findedElement = findFirstIndex elementToFind myList1

insertAtIndex :: Int -> a -> [a] -> [a]

insertAtIndex n element list = take n list ++ [element] ++ drop n list

insertPosition = 2

elementToInsert = 14

insertedList = insertAtIndex insertPosition elementToInsert myList2

countFrequency :: (Eq a) => [a] -> [(a, Int)]

countFrequency list = [(x, length (filter (== x) list)) | x <- nub list]

frequencyList = countFrequency myList3

convertToUpperCase :: String -> String

convertToUpperCase str = map toUpper str

inputString = "abcDf"

upperString = convertToUpperCase inputString

removeFirstOccurrence :: (Eq a) => a -> [a] -> [a]

removeFirstOccurrence element list = delete element list

elementToRemove = 3

removedList = removeFirstOccurrence elementToRemove myList3

splitPairs :: [(a, b)] -> ([a], [b])

splitPairs pairs = unzip pairs

myPairs = [(1, 'a'), (2, 'b'), (3, 'c')]

(firstComponents, secondComponents) = splitPairs myPairs

main :: IO ()

main = do

    print element

    print addedList

    print swapList

    print reversedList

    print findedElement

    print insertedList

    print frequencyList

    putStrLn upperString

    print removedList

    putStrLn "First Component:"

    print firstComponents

    putStrLn "Second Component:"

    print secondComponents

Висновок: Протягом виконання лабораторної роботи я навчився працювати з функціями для роботи з рядками