**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

**Лабораторна робота №2**

з дисципліни «Непроцедурне програмування»

Тема роботи: «Визначення рекурсивних функцій»

Варіант №9

Виконав студент

групи КН-32

Григор’єв Арсентій Георгійович

Перевірив:

Миколайчук Роман Антонович

**Київ – 2021**

**Мета роботи:** Набути досвiду визначення рекурсивних функцiй, використання механiзму зiставлення зi зразком i роботи з кортежами та списками.

**Завдання 1:** Визначте вказанi функцiї в кожному з завдань: а) без застосування, б) з застосуванням вбудованих функцiй.

****

Рисунок 1 – Завдання 1

Код:

|  |
| --- |
| import Data.List  --завдання 1  --a)  uniq2 :: [Int] -> [Int]  uniq2 [] = []  uniq2 (x:xs) = x: uniq2 (remove x xs)  where  remove :: Int -> [Int] -> [Int]  remove x [] = []  remove x (y:ys)  | x==y = remove x ys  | otherwise = y:remove x ys  --b)  uniq :: (Eq b, Ord b) => [b] -> [b]  uniq = map head.group.sort |

Результат роботи програми:

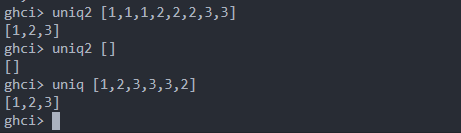


Рисунок 2 – результат роботи програми

**Завдання 2:**

****

Рисунок 3 – Завдання 2

Код:

|  |
| --- |
| --завдання 2  --а)  nsk :: Float -> Float -> Float  nsk 0 \_ = 0  nsk \_ 0 = 0  nsk x y = let gcd x y  | x > y = gcd (x-y) y  | x < y = gcd x (y-x)  | otherwise = x  in x\*y / gcd x y  --б)  lcm :: (Integral a) => a -> a -> a  lcm \_ 0 = 0  lcm 0 \_ = 0  lcm x y = abs (x \* y `quot` gcd x y ) |

Результат роботи програми:

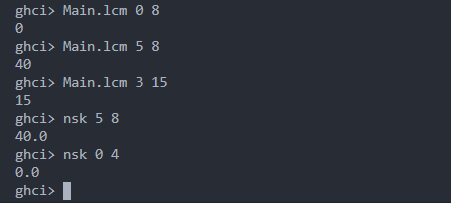


Рисунок 4 – результат роботи програми

**Висновок:** Під час лабораторної роботи ми ознайомились з реалізацією рекурсивних функцій та використанням вбудованих засобів мови Haskell. Також у нас була можливість попрацювати зі списками та кортежами. Порівнюючи рекурсію із вбудованими функціями, варто зазначити, що результат в обох випадках виявився однаковим, проте використання вбудованих функцій є зручнішим та більш читабельним.