ІП-11 Панченко Сергій Віталійович

Варіант 25 mod 15 = 10

Завдання

Визначте вказані функції в кожному з завдань: а) без застосування, б) з застосуванням вбудованих функцій (визначених у модулі Prelude).

1) Знайти передостанній елемент списку.
2) Циклічний правий зсув списку на n позицій.

Код

```
module Lib
    (
      oneTenA,
      oneTenB,
      twoTenA,
      twoTenB
    ) where

import Data.List

oneTenA :: [a] -> a
oneTenA [] = error "empty list"
oneTenA [_] = error "only one element"
oneTenA [x, _] = x
oneTenA (_:xs) = oneTenA xs

oneTenB :: [a] -> a
oneTenB xs = last (init xs)

myLast :: [a] -> a
myLast [] = error "empty list"
myLast [x] = x
myLast (_:xs) = myLast xs

myInit :: [a] -> [a]
myInit [] = error "empty list"
myInit [_] = []
myInit (x:xs) = x : myInit xs

twoTenA :: [a] -> Int -> [a]
twoTenA l 0 = l
twoTenA l n = twoTenA (myLast l : myInit l) (n - 1)

twoTenB :: [a] -> Int -> [a]
twoTenB l 0 = l
twoTenB l n = twoTenA (last l : init l) (n - 1)
```

Код тестів

```
import Lib


import Test.HUnit


-- Функція тестування для першого завдання.
-- Для зручності зробив вивід номера тестового варіанта, назву
createOneTestCases :: (Eq a, Show a) => String -> (t -> a) -> [(t, a)] -> [Test]
createOneTestCases baseName func testCases = [TestCase (assertEqual (baseName ++ " " ++ show i) (func x) y) | (i, (x, y)) <- zip [0..] testCases]


-- Тестові варіанти для першого завдання. Перший елемент кортежу - аргумент,
-- друге - очікуване значення тесту.
oneTenCases :: [([Int], Int)]
oneTenCases = [
    ([1..2], 1),
    ([1..3], 2),
    ([1..4], 3)
  ]


-- Створюємо тести для oneTestA
oneTenATest :: [Test]
oneTenATest = createOneTestCases "oneTenA" oneTenA oneTenCases


-- Створюємо тести для oneTestB
oneTenBTest :: [Test]
oneTenBTest = createOneTestCases "oneTenB" oneTenB oneTenCases


-- Тестові варіанти для другого завдання. Перший елемент кортежу -
-- це кортеж аргументів, другий - очікуване значення тесту.
```

```
twoTenCases :: [(([Int], Int), [Int])]
twoTenCases = [
    (([1, 2, 3, 4, 5], 0), [1, 2, 3, 4, 5]),
    (([1, 2, 3, 4, 5], 1), [5, 1, 2, 3, 4]),
    (([1, 2, 3, 4, 5], 2), [4, 5, 1, 2, 3]),
    (([1, 2, 3, 4, 5], 3), [3, 4, 5, 1, 2]),
    (([1, 2, 3, 4, 5], 4), [2, 3, 4, 5, 1]),
    (([1, 2, 3, 4, 5], 5), [1, 2, 3, 4, 5]),
    (([1, 2, 3, 4, 5], 6), [5, 1, 2, 3, 4]),
    (([1, 2, 3, 4, 5], 7), [4, 5, 1, 2, 3]),
    (([1, 2, 3, 4, 5], 8), [3, 4, 5, 1, 2]),
    (([1, 2, 3, 4, 5], 9), [2, 3, 4, 5, 1]),
    (([1, 2, 3, 4, 5], 10), [1, 2, 3, 4, 5])
  ]


-- Функція тестування для другого завдання.
createTwoTestCases :: (Eq a, Show a) => String -> (c -> b -> a) ->
[((c, b), a)] -> [Test]
createTwoTestCases baseName func testCases = [TestCase
(assertEqual (baseName ++ " " ++ show i) (func x y) z) | (i, ((x,
y), z)) <- zip [0..] testCases]


-- Створюємо тести для twoTenA
twoTenATest :: [Test]
twoTenATest = createTwoTestCases "twoTenA" twoTenA twoTenCases


-- Створюємо тести для twoTenB
twoTenBTest :: [Test]
twoTenBTest = createTwoTestCases "twoTenB" twoTenB twoTenCases


-- Створюємо тестовий список
tests :: Test
```

```
tests = TestList (oneTenATest ++ oneTenBTest ++ twoTenATest ++
twoTenBTest)


-- Головна функція, виводить результати тестування.
main :: IO ()
main  = do
    result <- runTestTT tests
    print (if failures result > 0 then "Failure" else "Success")
```

Результати тестування

```
usr/local/bin/stack test Lab1:test:Lab1-test --test-arguments --
color
Lab1-0.1.0.0: unregistering (local file changes:
.stack-work/dist/x86_64-linux-tinfo6/ghc-9.6.3/build/autogen/
Paths_Lab1.hs)
Lab1> build (lib + test)
Preprocessing library for Lab1-0.1.0.0..
Building library for Lab1-0.1.0.0..
Preprocessing test suite 'Lab1-test' for Lab1-0.1.0.0..
Building test suite 'Lab1-test' for Lab1-0.1.0.0..
Lab1> copy/register
Installing library in
/home/sideshowbobgot/university/FunctionalProgrammingHaskell/Labs/
Lab1/.stack-work/install/x86_64-linux-tinfo6/
bdcd8bafb08eb8fd3fc018c072ae11c880d48300c5c3f8101cdc772686b89541/9
.6.3/lib/x86_64-linux-ghc-9.6.3/Lab1-0.1.0.0-9vFcbDig3lKeAsPymFTvQ
Installing executable Lab1-exe in
/home/sideshowbobgot/university/FunctionalProgrammingHaskell/Labs/
Lab1/.stack-work/install/x86_64-linux-tinfo6/
bdcd8bafb08eb8fd3fc018c072ae11c880d48300c5c3f8101cdc772686b89541/9
.6.3/bin
Registering library for Lab1-0.1.0.0..
Lab1> test (suite: Lab1-test, args: --color)
```

```
Cases: 28  Tried: 28  Errors: 0  Failures: 0



Lab1> Test suite Lab1-test passed
Completed 2 action(s).
"Success"

Process finished with exit code 0
```