



Tugas 2

Tags

Identitas

- Nama: Fahdii Ajmalal Fikrie
- NPM: 1906398370

Soal 1

Buatlah fungsi `sumOfSquares` yang menerima sebuah list of integer dan mengembalikan sebuah integer yang merupakan penjumlahan dari kuadrat elemen list input!

```
listSumOfSquares :: [Int] -> Int
listSumOfSquares [] = 0
listSumOfSquares (x:t) = (x * x) + listSumOfSquares t

seqOfStringToInt :: String -> [Int]
seqOfStringToInt seq = map read $ words seq

main :: IO()
main = do
    seq <- getLine
    let listOfInt = seqOfStringToInt seq
    print $ listSumOfSquares listOfInt
```

Soal 2

Bilangan triangular adalah penjumlahan bilangan positif tersebut dengan seluruh bilangan bulat positif sebelumnya. Contohnya bilangan triangular ke-5 adalah $5+4+3+2+1$. Buatlah fungsi triangular yang menerima bilangan bulat positif n dan mengembalikan bilangan triangular yang ke- n !

```
{-
  Reference:
    > https://stackoverflow.com/questions/29943390/how-would-i-write-a-triangle-function-recursively
-}

triangular :: Int -> Int
triangular 0 = 0
triangular n = n + triangular (n-1)
```

```
main:: IO()
main = do
    putStrLn "Insert the nth number of triangular you want to count: "
    nth <- readLn
    print $ triangular nth
```

Soal 3

Buatlah fungsi power tanpa menggunakan fungsi pangkat yang sudah ada di Haskell, input dibatasi hanya untuk bilangan bulat positif

```
-- let power 0 f = (\x->x);
-- power n f = f.(power (n-1) f)
--     in (power 2 (\x -> x + 1)) 2

-- power base exp
--     | (exp == 0) = 1
--     | (exp > 0) = power base exp-1
--     | otherwise = 0

power x n =
    if n==0 then 1
    else if n>0 then x * power x (n-1)
    else error "negative power"

main :: IO()
main = do
    putStrLn "Insert base number: "
    base <- readLn :: IO Int
    putStrLn "Insert exponent number: "
    exp <- readLn
    print $ power base exp
```

Soal 4

Palindrome adalah kata yang dibaca sama dari depan ataupun belakang. Contohnya “Madam, I’m Adam”, “No lemon, no melon” dan lain-lain. Buatlah sebuah fungsi yang menerima string dan mengembalikan Boolean untuk mengecek apakah string tersebut palindrome atau tidak.

```
{-
    References:
    > https://stackoverflow.com/questions/30242668/remove-characters-from-string-in-haskell
    > https://codereview.stackexchange.com/questions/24340/palindrome-test-in-haskell
    > https://stackoverflow.com/questions/3098877/remove-white-space-from-string
-}

import Data.Char

isPalindrome :: String -> Bool
isPalindrome w = w == reverse w

stripLeadingWhitespace :: String -> String
stripLeadingWhitespace = unlines . map (dropWhile isSpace) . lines
```

```
removePunc :: String -> String
removePunc xs = [ x | x <- xs, not (x `elem` ",.?!-:;\"'\"") ]

main :: IO()
main = do
    putStrLn "Write a sentence that you'd like to do a palindrome check: "
    sentence <- getLine
    let cleanSentence = stripLeadingWhitespace removePunc sentence
    print $ isPalindrome cleanSentence
```

Refleksi



Saya masih belum mengerti beberapa hal di haskell, seperti:

- Typing (beda String, [Char], dst.)
- Function composition (terutama dalam membuat fungsi power)