საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ასოც.პროფ. ნათელ არჩვაძე

ქულათა ჯამი: 20

ქულათა გადანაწილების ინსტრუქცია: 1-4 საკითხი -2ქ., 5-8 საკითხი -3ქ. =20

1. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვს n-ს და აბრუნებს 1-დან n -მდე ნატურალური რიცხვის ფაქტორიალების სიას.

factorial 0 = 1 +2

factorial n = n \* factorial(n-1)

factorialsia 0 = [1]

factorialsia n = factorial n : factorialsia(n-1)

rfactorialsia n = reverse (factorialsia n)

1. სტანდარტული prelude ფაილის გამოუყენებლად რეკურსიის საშუალებით განსაზღვრეთ ფუნქცია, რომელიც დაადგენს, არის თუ არა მოცემული მნიშვნელობა სიის ელემენტი: elem::Eq a=>a->[a]->Bool.

iselem::Eq a=>a->[a]->Bool +2

iselem n [] = False

iselem n (x:xs)

| n == x = True

| otherwise = iselem n xs

1. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვებს a-ს და b-ს და აბრუნებს a-დან b -მდე ლუწი 11-ის ჯერადი რიცხვების სიას.

numbs1 a b x +2

| x > b = []

| x `mod` 11 == 0 && x `mod` 2 == 0 && x <= b = x : numbs1 a b (x+1)

| otherwise = numbs1 a b (x+1)

numbs a b = numbs1 a b a

1. განსაზღვრეთ ფუნქცია, რომელიც მოცემული სიიდან n-ურ წევრს შეცვლის მისი კვადრატით.

square \_ [] = [] +2

square 0 (x:xs) = x\*x : xs

square n (x:xs) = x : square (n-1) xs

1. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს სიას და აბრუნებს სიის ლუწი რიცხვების სიას გაერთიანებულს კენტი რიცხვების სიასთან თავდაპირველ სიაში მათი თანმიმდევრობის შენარჩუნებით. მაგალითად, fun [1,2,3,4,5,6,7,8,9] = [2,4,6,8, 1,3,5,7,9].

fun2 [] = [] +3

fun2 (x:xs)

| x `mod` 2 == 0 = x : fun2 xs

| otherwise = fun2 xs

fun1 [] = []

fun1 (x:xs)

| odd x = x : fun1 xs

| otherwise = fun1 xs

fun [] = []

fun (x:xs) = concat[fun2 (x:xs),fun1 (x:xs)]

1. ფუნქცია removeEmpty, რომელიც ამოაგდებს ცარიელ სტრიქონებს სტრიქონების მოცემული სიიდან. მაგალითად, removeEmpty ["", "Hello", "", "", "World!"] უნდა დააბრუნოს ["Hello","World!"].

removeEmpty :: [[Char]] -> [[Char]] +3

removeEmpty [] = []

removeEmpty (x:xs) = if x == "" then removeEmpty xs

else x : removeEmpty xs

1. განსაზღვრეთ ფუნქცია makePositive, რომელიც უცვლის ნიშანს რიცხვების სიის ყველა უარყოფით ელემენტს. მაგალითად, makePositive [-1, 0, 5, -10, -20] გვაძლევს [1,0,5,10,20].

makePositive :: [Integer]->[Integer] +3

makePositive [] = []

makePositive (x:xs) = if x < 0 then ((-1)\*x) : (makePositive xs)

else x : (makePositive xs)

1. განსაზღვრეთ ფუნქცია, რომელიც მოცემული არგუმენტიდან (სიიდან) ამოაგდებს მხოლოდ ორნიშნა რიცხვებს.

remove2 :: [Integer]->[Integer] +3

remove2 [] = []

remove2 (x:xs) = if x >= 10 && x <= 99 then remove2 xs

else x : remove2 xs