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. =17

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

nSquare :: (Eq a, Num a) => a -> [a] +2

nSquare 0 = []

nSquare n = n\*n : nSquare (n-1)

result n = reverse (nSquare n)

1. სტანდარტული prelude ფაილის გამოუყენებლად რეკურსიის საშუალებით განსაზღვრეთ ფუნქცია, რომელიც შექმნის n ერთნაირი ელემენტების სიას: replicate::Int->a->[a].

rep 0 x = [] +2

rep n x = x : rep (n-1) x

1. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვებს n-ს და m-ს და აბრუნებს n-დან m -მდე კენტი სამის ჯერადი რიცხვების სიას.

funcf:: Integral a => a -> a -> [a]

funcf n m = sq n m n

funcx :: Integral a => t -> a -> a -> [a]

funcx n m b

| b > m = []

| mod b 3 == 0 && odd b = b : sq n m (b+6)

| otherwise = sq n m (b+1)

1. ფუნქცია ამოაგდებს მოცემული სიიდან n - ურ წევრს. შეცვალეთ ფუნქციის განმარტება - გაითვალისწინეთ შემთხვევა, როცა n-ის მნიშვნელობა სიის სიგრძეზე მეტია.

func :: (Eq t, Num t) => t -> [a] -> [a]

func \_ [] = errorBadArgument +2

func 0 (x:xs) = xs

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

1. ორი სიის ელემენტების აჯამვის ფუნქცია. აბრუნებს სიას, რომელიც შედგება პარამეტრი სიების ელემენტების ჯამისგან. გაითვალისწინეთ, რომ გადაცემული სიები შეიძლება იყოს სხვადასხვა სიგრძის. მაგალითად, listsum [1,2,3][4,5]=[5,7,3].

listsum :: Num a => [a] -> [a] -> [a] +3

listsum [] [] = []

listsum [] xs = xs

listsum xs [] = xs

listsum (x:xs) (y:ys) = x+y : listsum xs ys

1. ფუნქცია substitute :: Char -> Char -> String -> String, რომელიც ცვლის მოცემულ სიმბოლოს მეორე სიმბოლოთი. მაგალითად, substitute ’e’ ’i’ "eigenvalue" აბრუნებს "iiginvalui".

substitute :: Char -> Char -> String -> String +3

substitute a b [] = []

substitute a b (x:xs)

| a == x = b : substitute a b xs

| otherwise = x : substitute a b xs

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

makePositive :: (Ord a, Num a) => [a] -> [a] +2

makePositive [] = []

makePositive (x:xs)

| x < 0 = negate x : makePositive xs

|otherwise = x : makePositive xs

1. განსაზღვრეთ ფუნქცია, რომელიც მოცემული არგუმენტიდან (სიიდან) ამოაგდებს მხოლოდ იმ a რიცხვებს, რომლებიც აკმაყოფილებენ პირობას: 150<a<400.

funcCut :: (Ord a, Num a) => [a] -> [a] +3

funcCut [] = []

funcCut (x:xs)

| x > 150 && x < 400 = funcCut xs

|otherwise = x : funcCut xs