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. =16

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

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

squareList 0 = []

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

squareListn = reverse (squareList n)

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

**სიის კონსტრუქტორით**: func2 n x = [x | k <- [1..n]] +2

**რეკურსიულად:**  func2 0 x = []

func2 n x = x : myfunc (n-1) x ??????????

func2 :: Int -> a-> [a]

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

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

funcListOdd[] = [] +2-

funcListOdd(x:xs)

| mod b 3 == 0 && odd x = x : funcListOdd xs

| otherwise = funcListOdd xs

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

delete :: Eq a => a -> [a] -> [a] +1

delete n [] = []

delete n (x:xs)

| x == n = delete n xs

|otherwise = x : delete n 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].
2. 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] +3

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.

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

removeItem [] = []

removeItem (x:xs)

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

|otherwise = x : removeItem xs