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. =9

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

sia :: (Eq a, Num a) => a -> [a] +1

sia 0 = []

sia n = n : sia (n-1)

bolosia a = reverse (sia a)

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

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

chemielem n [] = False

chemielem n (x:xs)

| n == x = True

| otherwise = chemielem 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.
2. განსაზღვრეთ ფუნქცია, რომელიც მოცემული სიიდან n-ურ წევრს შეცვლის მისი კვადრატით.

fun\_1\_d 0 = [] -

fun\_1\_d n = n^2:fun\_1\_d(n-1)

fun\_1\_dr n = reverse(fun\_1\_d n)

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].

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

swaplist [] = []

swaplist (x:[]) = []

swaplist (x:y:xs)

| odd x && even y = y:x : swaplist xs

| even x && odd y = y:x : swaplist xs

| otherwise = swaplist xs

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

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

washla [] = []

washla (x:xs)

| x > 9 && x < 100 = washla xs

|otherwise = x : washla xs