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. =4

1. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვს n-ს და აბრუნებს 1-დან n -მდე ნატურალური რიცხვის ფაქტორიალების სიას.
2. სტანდარტული prelude ფაილის გამოუყენებლად რეკურსიის საშუალებით განსაზღვრეთ ფუნქცია, რომელიც დაადგენს, არის თუ არა მოცემული მნიშვნელობა სიის ელემენტი: elem::Eq a=>a->[a]->Bool.
3. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვებს a-ს და b-ს და აბრუნებს a-დან b -მდე ლუწი 11-ის ჯერადი რიცხვების სიას.
4. განსაზღვრეთ ფუნქცია, რომელიც მოცემული სიიდან n-ურ წევრს შეცვლის მისი კვადრატით.
5. 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].
6. ფუნქცია removeEmpty, რომელიც ამოაგდებს ცარიელ სტრიქონებს სტრიქონების მოცემული სიიდან. მაგალითად, removeEmpty ["", "Hello", "", "", "World!"] უნდა დააბრუნოს ["Hello","World!"].
7. განსაზღვრეთ ფუნქცია makePositive, რომელიც უცვლის ნიშანს რიცხვების სიის ყველა უარყოფით ელემენტს. მაგალითად, makePositive [-1, 0, 5, -10, -20] გვაძლევს [1,0,5,10,20].
8. განსაზღვრეთ ფუნქცია, რომელიც მოცემული არგუმენტიდან (სიიდან) ამოაგდებს მხოლოდ ორნიშნა რიცხვებს.

1.

factorial :: (Eq p, Num p) => p -> p +2

factorial 0 = 1

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

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

fctrl 0 = [1]

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

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

myfctrl n = reverse (fctrl n)

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

fctrlmore [] = []

fctrlmore (x:xs) = factorial x : fctrlmore xs

2.

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

elmnt n [] = False

elmnt n (x:xs)

    | n == x = True

    | otherwise = elmnt n xs

3.