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;
2. სტანდარტული prelude ფაილის გამოუყენებლად რეკურსიის საშუალებით განსაზღვრეთ ფუნქცია, რომელიც შექმნის n ერთნაირი ელემენტების სიას: replicate::Int->a->[a].
3. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვებს n-ს და m-ს და აბრუნებს n-დან m -მდე კენტი სამის ჯერადი რიცხვების სიას.
4. ფუნქცია ამოაგდებს მოცემული სიიდან n - ურ წევრს. შეცვალეთ ფუნქციის განმარტება - გაითვალისწინეთ შემთხვევა, როცა n-ის მნიშვნელობა სიის სიგრძეზე მეტია.
5. 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].
6. ფუნქცია substitute :: Char -> Char -> String -> String, რომელიც ცვლის მოცემულ სიმბოლოს მეორე სიმბოლოთი. მაგალითად, substitute ’e’ ’i’ "eigenvalue" აბრუნებს "iiginvalui".
7. განსაზღვრეთ ფუნქცია makePositive, რომელიც უცვლის ნიშანს რიცხვების სიის ყველა ელემენტს. მაგალითად, makePositive [-1, 10, 5, -10, -20] გვაძლევს [1,-10,-5,10,20].
8. განსაზღვრეთ ფუნქცია, რომელიც მოცემული არგუმენტიდან (სიიდან) ამოაგდებს მხოლოდ იმ a რიცხვებს, რომლებიც აკმაყოფილებენ პირობას: 150<a<400.

1. 1-დან n -მდე ნატურალური რიცხვის კვადრატების სიას; +2

ksia 0 = []

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

rksia n = reverse (ksia n)

2. siaa 0 x = []

siaa n x = x : (siaa (n-1) x) +2

1. 5. listsum :: Num a => [a] -> [a] -> [a] listsum [] [] = [] listsum [] xs = xs listsum xs [] = xs listsum (x:xs) (y:ys) = x+y : listsum xs ys +2
2. 6. removeEmpty :: [[Char]] -> [[Char]] removeEmpty [] = [] removeEmpty (x:xs) | x == "" = removeEmpty xs |otherwise = x : removeEmpty xs +3
3. 7. makePositive :: (Ord a, Num a) => [a] -> [a] makePositive [] = [] makePositive (x:xs) | x < 0 = negate x : makePositive xs |otherwise = x : makePositive x +3
4. 8. amogdeba :: (Ord a, Num a) => [a] -> [a] amogdeba [] = [] amogdeba (x:xs) | x > 150 && x < 400 = amogdeba xs |otherwise = x : amogdeba xs +3
5. 3. kk :: Integral a => a -> a -> [a] kk n m = gg n m n jj :: Integral a => t -> a -> a -> [a] jj n m b | b > m = [] | mod b 3 == 0 && odd b = b : gg n m (b+6) | otherwise = gg n m (b+1) +2