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. =19

1. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვს n-ს და აბრუნებს 1-დან n -მდე 2-ის ხარისხების სიას.+2
2. getPowerOfTwo :: Int -> [Int]
3. getPowerOfTwo n | n < 1 = []
4. | otherwise = concat [(getPowerOfTwo (n - 1)), [2 ^ n]]
5. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვს n-ს და სიმბოლოს, და აბრუნებს სტრიქონს, რომელშიც სიმბოლო n-ჯერ არის განმეორებული. მაგალითად, ff 5 '%'="%%%%%" +2
6. repeatSymbol :: Int -> Char -> String
7. repeatSymbol n c | n < 1 = ""
8. | otherwise = [c] ++ (repeatSymbol (n - 1) c)
9. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვებს a-ს და b-ს და აბრუნებს a-დან b -მდე ლუწი ხუთის ჯერადი რიცხვების სიას. +2
10. getListOfTen :: Int -> Int -> [Int]
11. getListOfTen a b | a > b = []
12. | mod a 10 == 0 = concat[[a], (getListOfTen (a + 1) b)]
13. | otherwise = getListOfTen (a + 1) b
14. ფუნქცია გამოყოფს მოცემული სიის n - ურ წევრს. შეცვალეთ ფუნქციის განმარტება - გაითვალისწინეთ შემთხვევა, როცა n-ის მნიშვნელობა სიის სიგრძეზე მეტია. +1
15. getListPrefix :: Int -> [Int] -> [Int]
16. getListPrefix n [] = []
17. getListPrefix n (x:xs) | n < 1 = []
18. | (length xs) == 0 = [x]
19. | otherwise = concat [[x], (getListPrefix (n -1 ) xs)]
20. განსაზღვრეთ ორი სიის ელემენტების აჯამვის ფუნქცია. აბრუნებს სიას, რომელიც შედგება პარამეტრი სიების ელემენტების ჯამისგან. თუ არგუმენტებს სხვადასხვა სიგრძე აქვთ, ფუნქციის შედეგი იყოს უფრო მოკლე სიის სიგრძის. მაგალითად, listsum [1,2,3,8,3][4,5]=[5,7]; listsum [8][1,2,3,4,5]=[9] +3
21. listSum :: [Int] -> [Int] -> [Int]
22. listSum [] b = []
23. listSum a [] = []
24. listSum [] [] = []
25. listSum (a:as) (b:bs) = concat [[a + b], (listSum as bs)]
26. ფუნქცია, რომელიც აადგილებს მოცემულ სიაში მეზობელ ლუწ და კენტ ადგილას მყოფ ელემენტებს ელემენტებს.+3
27. swapElements :: [Int] -> [Int]
28. swapElements [] = []
29. swapElements list | (length list) == 1 = list
30. | (length list) == 2 = [(list!!1), (list!!0)]
31. | otherwise = concat [[(list!!1), (list!!0)], (swapElements (snd (splitAt 2 list)))]
32. ფუნქცია delete :: Char -> String -> String, რომელიც იღებს შესასვლელზე სტრიქონს და სიმბოლოს და აბრუნებს სტრიქონს, რომლიდანაც ამოშლილია ყველა სიმბოლო გარდა მოცემული სიმბოლოსი. მაგალითად, delete ’l’ "Hello world!" უნდა დააბრუნოს "ll". +3
33. keepChar :: Char -> String -> String
34. keepChar c "" = ""
35. keepChar c (x:xs) | x == c = [x] ++ (keepChar c xs)
36. | otherwise = (keepChar c xs)
37. განსაზღვრეთ ფუნქცია, რომელიც მოცემული არგუმენტიში (სიაში) დატოვებს მხოლოდ იმ a რიცხვებს, რომლებიც აკმაყოფილებენ პირობას: 150<a<400. +3
38. keepNumber :: [Int] -> [Int]
39. keepNumber [] = []
40. keepNumber (a:as) | (a > 150) && (a < 400) = concat [[a], (keepNumber as)]
41. | otherwise = keepNumber as