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. =15

1. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვს n-ს და აბრუნებს 1-დან n -მდე 2-ის ხარისხების სიას.

პასუხი:

powerOfTwo :: Int -> [Int] +2

powerOfTwo n = map (^2) [0..n]

1. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვს n-ს და სიმბოლოს, და აბრუნებს სტრიქონს, რომელშიც სიმბოლო n-ჯერ არის განმეორებული. მაგალითად, ff 5 '%'="%%%%%"

პასუხი:

ff :: Int -> Char -> String +2

ff n x = replicate n x

1. განსაზღვრეთ ფუნქცია, რომელიც შესასვლელზე ღებულობს მთელ რიცხვებს a-ს და b-ს და აბრუნებს a-დან b -მდე ლუწი ხუთის ჯერადი რიცხვების სიას.

პასუხი:

multiplyOf5FromRange :: Int -> Int -> [Int] +1

multiplyOf5FromRange a b = filter (statement) [a..b]

where statement x = mod x 5 == 0

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

პასუხი:

findNth :: Int -> [a] -> a +2

findNth \_ [] = error "List is Empty"

findNth y (x:xs) | y <= 0 = x

| otherwise = findNth (y-1) xs

1. 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]

პასუხი:

listsum :: [Int] -> [Int] -> [Int] +2

listsum lst1 lst2 = begin lst1 lst2 []

begin [] lst2 sumofList = (sumofList++lst2)

begin lst1 [] sumofList = (sumofList++lst1)

begin lst1 lst2 sumofList = begin (tail(lst1)) (tail(lst2)) (sumofList++[(head(lst1))+(head(lst2))])

1. ფუნქცია, რომელიც აადგილებს მოცემულ სიაში მეზობელ ლუწ და კენტ ადგილას მყოფ ელემენტებს ელემენტებს.

პასუხი:

–– ადგილების გაცვლა მეზობელი ლუწი და კენტ ადგილზე მყოფი ელემენტებისთვის

swapOddEven :: [Int] -> [Int] -

swapOddEven [] = []

swapOddEven (x:y:rest) = if (checkOddEven x y)==1 then

y:x:( swapOddEven rest) else

x:y:( swapOddEven rest)

swapOddEven [x] = [x]

–– ლუწ–კენტობის შემოწმება

checkOddEven :: Int -> Int -> Int

checkOddEven x y = if (mod x 2)==1 then

if (mod y 2)==0 then 1 else 0 მკაცრად ტიპიზირებული ენაა!!!

else if (mod x 2)==0 then

if (mod y 2)==1 then 1 else 0

else 0

არ მუშაობს swapOddEven [2,4,1,3]

[2,4,1,3]

1. 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 "lll".

პასუხი:

delete :: Char -> String -> String +3

delete \_ [] = []

delete n (x:xs) = if(x /= n) then delete n xs

else x: delete n xs

1. განსაზღვრეთ ფუნქცია, რომელიც მოცემული არგუმენტიში (სიაში) დატოვებს მხოლოდ იმ a რიცხვებს, რომლებიც აკმაყოფილებენ პირობას: 150<a<400.

პასუხი:

findRangeNumbersOnly :: [Int] -> [Int] +3

findRangeNumbersOnly [] = []

findRangeNumbersOnly x = filter (< 400) (filter (> 150) x)