საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25=5+4+4+5+5

ქულათა გადანაწილების ინსტრუქცია: 5 საკითხი, თითოეული 5 ქულა.

ბილეთი =23

1. განსაზღვრეთ შემდეგი ფუნქცია ორი ხერხით: პირობითი ოპერატორისა და დაცული განტოლებების გამოყენებით:  მნიშვნელობის გამოსათვლელად. როგორ მიმართავთ ამ ფუნქციას? ფუნქციის ტიპი განსაზღვრეთ ცხადად.

fun x

| x == -2.5 = 3.75\*x^2+(10/(2-x)\*(1+4\*x))

| x> (-3) && x<7 = ((1-x^2+8.2\*x)/(x+2.5))-4

| otherwise = 5\*x-7 + (3.5\* x^3)/(4\*(1.5+x))  
  
fun x = if x == -2.5 then 3.75\*x^2+(10/(2-x)\*(1+4\*x)) else if x> -3 && x<7 then ((1-x^2+8.2\*x)/(x+2.5))-4 else 5\*x-7 + (3.5\* x^3)/(4\*(1.5+x))  
  
fun 1235

fun :: (Ord a, Fractional a) => a -> a

1. განსაზღვრეთ შემდეგი ფუნქციები λ ნოტაციის გამოყენებით და მიუთითეთ შესაბამისი ტიპები:

* ფუნქცია, რომელიც აბრუნებს სიის მეორე ელემენტს.   
  let second = \x -> tail(take 2 x) +2  
  second :: [a] -> [a]
* ფუნქცია, რომელიც გადაცემული სიისთვის აბრუნებს ლუწი ელემენტების კვადრატებს, კენტების -კუბებს.   
    
  first [] = []  
  first (x: xs)= x : second xs  
    
  second []= []   
  second (x:xs) = first xs  
    
  fun = \x -> (map (\y->y^2) (second x) , map (\y-> y^3) (first x))  
    
  fun :: [Integer] -> ([Integer], [Integer]) სხვა შინაარსის ამოცანაა -
* ფუნქცია, რომელსაც გადაეცემა ელემენტი და სია და დააბრუნებს ახალ სიას, სადაც ბოლო ელემენტად ჩასმულია პირველი არგუმენტი.  
    
  fun a [] = [a] +2  
  fun a (x:xs) = x : fun a xs  
  fun :: t -> [t] -> [t]

1. განსაზღვრეთ ცხრილით მოცემული $ ფუნქცია **რამდენიმე** შაბლონის გამოყენებით:

|  |  |  |
| --- | --- | --- |
| a | b | a $$ b |
| False | False | True |
| False | True | True |
| True | False | False |
| True | True | True |

განსაზღვრეთ ფუნქციის ტიპი.  
($) False False = True

($) False True = True

($) True False = False

($) True True = True

($) :: Bool -> Bool -> Bool

(>>=) False False = True ????? იგივეა -

(>>=) False True = True

(>>=) True False = False

(>>=) True True = True

(>>=) :: Bool -> Bool -> Bool

1. სიის კონსტრუქტორის გამოყენებით შეადგინეთ:

* იმ სამნიშნა რიცხვების სია, რომელთა ციფრთა ჯამი არის კენტი რიცხვი.  
  [x | x<- [100..999] , (x `div` 100 + (x `div` 10) `mod` 10 + x`mod` 10) `mod` 2 == 1]
* შეადგინეთ ისეთი (x,y,z) სამეულების სია, სადაც თითოეული ცვლადი არის მარტივი, მნიშვნელობით 1–დან n-მდე და ჭეშმარიტია პირობა x+y==z.  
  let factors n = [x | x <- [1..n], mod n x == 0]  
  let prime n = factors n == [1,n]  
  let fun n = [(x,y,z) | x<-[1..n], y<-[1..n] , z<-[1..n], prime x , prime y ,prime z, x+y==z ]
* სია [22, 33, ... 2020].   
  [x^x | x <- [2..20]]

1. განსაზღვრეთ ფუნქცია ორი ხერხით: REPL გარემოში და საწყისი კოდის სახით. ფუნქცია გამოიძახეთ კონკრეტული მონაცემებისთვის:

* ფუნქცია, რომელიც აწყვილებს თავის ორ არგუმენტს.  
  let pair x y = (x,y)  
  pair x y = (x,y)
* ფუნქცია, რომლის არგუმენტია სტრიქონი და რომელიც ამოწმებს, არის თუ არა არგუმენტი პალინდრომი.  
  let palindrome x = x == reverse x  
  palindrome x = x == reverse x
* ფუნქცია isSorted, რომელიც შესასვლელზე ღებულობს სამ რიცხვს და აბრუნებს True, თუ ეს რიცხვები დალაგებულია ზრდადობით ან კლებადობით.  
    
  isSorted x y z

| (x <= y && x <= z && y<= z) || (x >= y && x >= z && y >= z)= True

| otherwise = False

isSorted x y z = if (x <= y && x <= z && y<= z) || (x >= y && x >= z && y >= z) then True else False