საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25=4+5+5+5+5

ქულათა გადანაწილების ინსტრუქცია: 5 საკითხი, თითოეული 5 ქულა.

ბილეთი=24

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

1. y x = if x>5 then (4\*x-5\*x^2+2)/(2+2\*x)-1/(3\*(x-1)) else if x<=5 && x /=(-8) then 3\*x^2-5/2\*(x+8)+10 else (7\*x+1/(1+x))/(x^2-1)+1/2\*x

ამ ფუნქციის ტიპია : (Ord a, Fractional a) => a -> a

გამოძახება :

**\*Test>** y 1

-9.5

it :: Double

**\*Test>** y 2

-3.0

it :: Double

დაც. განტ -1

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

* ფუნქცია, რომელიც აბრუნებს გაორმაგებულ არგუმენტს.
  + პას : (\x->2\*x)

მაგალითად :

**\*Test>** (\x->2\*x) 6

12

ტიპი : Num a => a -> a

* ფუნქცია, რომელსაც გადაეცემა ორი არგუმენტი და ითვლის მთელ ხარისხს.
  + პას : (\x y->x^y)

მაგალითად :

**\*Test>** (\x y->x^y) 2 3

8

ტიპი : (Num a, Integral b) => a -> b -> a

* ფუნქცია, რომელსაც გადაეცემა ორი წყვილი და დააბრუნებს შესაბამისი ელემენტების ჯამების წყვილს. მაგალითად, არგუმენტებია: (1,2), (3,4), შედეგია: (4,6).
  + პას : (\(a,b) (c,d)->(a+c,b+d))

მაგალითად :

**\*Test>** (\(a,b) (c,d)->(a+c,b+d)) (1,2) (3,4)

(4,6)

ტიპი : (Num t1, Num t) => (t, t1) -> (t, t1) -> (t, t1)

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.

მაგალითად :

func True False = False

func \_ \_ = True

ტიპი : Bool -> Bool -> Bool

ან :

func1 False \_ = True

func1 True True = True

func1 True False = False

1. სიის კონსტრუქტორის გამოყენებით შეადგინეთ:

* იმ სამნიშნა რიცხვების სია, რომელთა ციფრთა ჯამი არის 3-ის ჯერადი რიცხვი;

პასუხი : [a|a<-[100..999], (a`mod`10+ a`div`10`mod`10+a`div`100)`mod`3==0]

* მოცემულ n რიცხვამდე ისეთი რიცხვების სია, რომელიც ორი მარტივი რიცხვის ჯამით; წარმოდგება.

მაგალითად, როცა n=5

let n = 5

[(x+y)| x<-[1..n],y<-[1..n], prime x,prime y]

(

factors n = [x | x<-[1..n], n `mod` x == 0]

prime n = factors n == [1,n]

)

* გამოსახულება , რომელიც პირველი 100 რიცხვის 12 + 22 + . . . + 1002 ჯამს ანგარიშობს.

**\*Test>** sum[(x^2)|x<-[1..100]]

338350

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

* ფუნქცია, რომელიც მოცემულ წყვილში ადგილებს უცვლის ელემენტებს.
  + 1. func (x,y)=(y,x)

**\*Test>** func (3,4)

(4,3)

* + 2.

**\*Test>** (\(x,y)->(y,x)) (5,6)

* + (6,5)
* ფუნქცია, რომელიც გადაცემული რიცხვების სიისთვის ამოწმებს თითოეული ელემენტის ლუწ-კენტობას და გვიბრუნებს ლოგიკური მნიშვნელობების სიას.
  + მაგალითად :

**\*Test>** map(\x->x`mod`2==0) [2,3,4,7,8,9]

[True,False,True,False,True,False]

* ფუნქცია isTriangle, რომელიც განსაზღვრავს, შეიძლება თუ არა მოცემულ x, y და z სიგრძის მონაკვეთებზე აიგოს სამკუთხედი.
  + 1. isTriangle a b c = a+b>c && a+c>b && b+c>a

**\*Test>** isTriangle 3 4 5

True

it :: Bool

**\*Test>** isTriangle 3 4 9

False

it :: Bool

2.

**\*Test>** (\a b c -> a+b>c && a+c>b && b+c>a) 4 5 6

True