**კურსის დასახელება:** ფუნქციონალური დაპროგრამება **სტატუსი:** ძირითადი

**ქულების განაწილება:** 1 - **5**ქ., 2 - **5** ქ., 3 - **5** ქ., 4 - **5** ქ., 5 - 1**5** ქ. =32 **ქულათა ჯამი:** **40** ქულა

**გამოცდის ხანგრძლივობა: 2 საათი.**

**სტუდენტი: დოლიძე ნანი**

**ბილეთი**

1. სიის კონსტრუქტორის გამოყენებით დაწერეთ mymap სახელწოდების მაღალი რიგის საბიბლიოთეკო ფუნქციის განმარტების თქვენი ვერსია, რომელიც ახორციელებს მითითებულ მოქმედებას (ფუნქციას) სიის ყოველ ელემენტზე . განსაზღვრეთ ფუნქციის ტიპი. მოიყვანეთ გამოძახების მაგალითი.

1)mymap f xs =[ f x | x <- xs] ფუნქციის ტიპი იქნება: mymap :: (t -> a) -> [t] -> [a] +5

ფუნქციის გამოძახების მაგალითი:

mymap (\*2) [1,2,3,4,5]

დააბრუნებს:

[2, 4, 6, 8, 10]

1. რეკურსიის საშუალებით. დაწერეთ მაღალი რიგის საბიბლიოთეკო myfilter ფუნქცია, რომელიც ირჩევს სიიდან ყოველ ელემენტს, რომელიც პრედიკატს აკმაყოფილებს განსაზღვრეთ ფუნქციის ტიპი. მოიყვანეთ გამოძახების მაგალითი.

2) myfilter pred xs = [x | x <- xs, pred x] +5

ფუნქციის ტიპი იქნება: myfilter :: (a -> Bool) -> [a] -> [a]

გამოძახების მაგალითი:

myfilter odd [1,2,3,4,5]

დააბრუნებს:

[1, 3, 5]

1. განსაზღვრეთ რეკურსიულად ან სიის კონსტრუქტორის გამოყენებით ფუნქცია, რომელიც შესასვლელზე ღებულობს ნამდვილ რიცხვების სიას და ითვლის ორ მნიშვნელობას: მათ საშუალო არითმეტიკულს და საშუალო გეომეტრიულს. შეეცადეთ, რომ ფუნქციამ მხოლოდ ერთხელ გადახედოს სიას. განსაზღვრეთ ფუნქციის ტიპი. მოიყვანეთ გამოძახების მაგალითი.

3) avarageFunction :: (Num a) => [a] -> a +2

avarageFunction [ ] = 0

avarageFunction xs = sum xs / length xs

1. განსაზღვრეთ maRali rigis *any* da *takeWhile* funqciebi standartul prelude failSi mocemuli maTi aRwerebis gamouyeneblad.

4) pred [ ] = False any ????????

any pred (x: xs)

| pred x = True

|otherwise any pred xs

takeWhile pred [] = [] +3

takeWhile pred (x: xs)

| pred x = x : takeWhile pred xs

|otherwise = []

1. უძრავი ქონების სააგენტოში იყიდება ბინები- Flat, ოთახები- Room და კერძო სახლები-House. ბინა ხასიათდება სართულით, ფართობით და სახლის სართულების რაოდენობით. ოთახი ხასიათდება ამის გარდა კიდევ ფართობით (დამატებით მთელი ბინის ფართობისა). კერძო სახლი ხასიათდება მხოლოდ ფართობით. ანუ განსაზღვრულია მონაცემთა ტიპი, რომელიც წარმოადგენს უძრავი ქონების ობიექტებზე ინფორმაციას. ანუ მოცემული გვაქვს ტიპი:

data NedvObject = Flat Int Int Int | Room Int Int Int Int | House Int deriving (Eq,Show)

-- Flat sarTuli, farTobi, sarTulianoba | Room sarTuli, farTobi, sarTulianob,a oTaxis farTobi | House ფართობი

მონაცემთა ბაზაში ინახება მნიშვნელობების წყვილები, რომელთაგან პირველი წარმოადგენს უძრავ ობიექტს, მეორე–მის ფასს.

განსაზღვრეთ შემდეგი ფუნქციები:

1. GetRoom - მონაცემთა ბაზიდან არჩევს ოთახებს;
2. getFlatByPrice - ბაზიდან ირჩევს ბინებს, რომელთა ფასი მოცემულის ტოლია;
3. getExceptBounds, irCevs binebs mocemul sarTulze;
4. getByType - მეორე არგუმენტად გადაეცემა სტრიქონი, ფასი და ბაზიდან შესაბამის ინფორმაციას იღებს. მაგალითად,

**\*Main>** getByType [(Flat 3 100 10,1000),(Room 4 120 9 20,1000),(House 200,1000),(Flat 1 100 10,900)] "Room" 1001

[(Room 4 120 9 20,1000)]

it :: [(NedvObject, Int)]

it :: [(NedvObject, Int)]

1)GetRoom +17

GetRoom :: [(NedvObject,Int)] -> [(NedvObject,Int)]

GetRoom [] = []

GetRoom ((Room a b c d,y):xs) = (Room a b c d,y):GetRoom xs

GetRoom (\_:xs) = GetRoom xs

2)getFlatByPrice

getFlatByPrice (Flat a b c, y): xs) price =

if y<price then (Flat a b c, y): getFlatByPrice xs price

else getFlatByPrice xs price

3)getExceptBounds

getExceptBounds:: [(NedvObject,Int)] -> [(NedvObject,Int)]

getExceptBounds [] = []

getExceptBounds ((Flat a b c, y): xs) =

if (a/=c) then (Flat a b c, y) : getExceptBounds xs

else getExceptBounds xs

4)getByType :: [(NedvObject, Int)] -> String -> [(NedvObject, Int)]

getByType [] \_ =[]

getByType (x: xs) type = case x of

(Flat a b c, y) -> if type==”Flat” then x:getByType xs type

else x:getByType xs type

(Room a b c d, y) -> if type==”Room” then x:getByType xs type

else x:getByType xs type

(House a , y) -> if type==”House” then x:getByType xs type

else x:getByType xs type