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

**ქულების განაწილება:** 1 - **5**ქ., 2 - **5** ქ., 3 - **5** ქ., 4 - **5** ქ., 5 - 20 ქ. = 36 **ქულათა ჯამი:** **40** ქულა

**გამოცდის ხანგრძლივობა: 2 საათი.**

**ბილეთი**

1. რეკურსიის 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.

mymap :: (a-> b) -> [a] -> [b] +5

|  |  |
| --- | --- |
|  | mymap f [] = [] |
|  | mymap f (x:xs) = f x :mymap f xs |
| mymap(+5)([1..5]::[Int]) |  |

[6,7,8,9,10]

ფუნქციის ტიპი:(a-> b) -> [a] -> [b]

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

myfilter :: (a -> Bool) -> [a] -> [a] +5

myfilter f xs = [x | x <- xs, f x]

myfilter (>= 3) [1..10] :: [Int] [3,4,5,6,7,8,9,10]

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

1. განსაზღვრეთ რეკურსიულად და სიის კონსტრუქტორის გამოყენებით ფუნქცია, რომელიც შესასვლელზე ღებულობს სიას და აბრუნებს ორ მნიშვნელობას: თავდაპირველი სიიდან ლუწი რიცხვების სიას გაერთიანებულს კენტი რიცხვების სიასთან და დადებითი რიცხვების სიას გაერთიანებულს კენტი რიცხვების სიასთან თავდაპირველ სიაში მათი თანმიმდევრობის შენარჩუნებით. განსაზღვრეთ ფუნქციის ტიპი და მოიყვანეთ გამოძახების მაგალითი.

fun3 odd [] = []

fun3 odd (x:xs)  +2

|x `mod` 2 ==1 =x:fun3 odd xs

|otherwise= fun3 odd xs

fun3 even [] = []

fun3 even (x:xs)

|x `mod` 2==0 =x:fun3 even xs

|otherwise =fun3 even xs

fun3 xs= fun3odd  xs++ fun3 even xs

1. gansazRvreT maRali rigis *all* da *dropWhile* funqciebi standartul prelude failSi mocemuli maTi aRwerebis gamouyeneblad.

all' t [] = True +4

all' t (x:xs)

| not (t x) = False

| otherwise = all' t xs

**\*Main>** all (<11) [1..10]

True

it :: Bool

dropWhile \_ [] = []

  dropWhile p (x : xs)

  | p x  = dropWhile p xs

| otherwise = x : xs

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: +20

1. getFlat monacemTa bazidan irCevs binebs;

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

getFlat [] = []

getFlat ((Flat x t r,y):xs) = (Flat x t r,y): getFlat xs

getFlat (\_:xs) = getFlat xs

გამოძახების მაგალითი:

**\*Main>** getFlat [(Flat 3 100 10,1000),(Room 4 120 9 20,500),(House 200,2000),(Flat 1 100 10,900)]

[(Flat 3 100 10,1000),(Flat 1 100 10,900)]

it :: [(NedvObject, Int)]

1. getRoomByPrice - ბაზიდან ირჩევს ოთახებს, რომელთა ფასი მოცემულზე მეტია;

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

getRoomByPrice [] \_ = []

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

        if y>price then (Room a b c d,y): getRoomByPrice xs price

        else getRoomByPrice xs price

გამოძახების მაგალითი:

**\*Main>** getRoomByPrice [(Room 4 123 9 20,600),(Room 4 124 9 20,700),(Room 4 120 9 20,800),(Room 4 156 9 20,900)] 650

[(Room 4 124 9 20,700),(Room 4 120 9 20,800),(Room 4 156 9 20,900)]

it :: [(NedvObject, Int)]

1. getExceptBounds, ირჩევს მონაცემთა ბაზიდან ბინებს, რომლებიც არ მდებარეობს პირველ და ბოლო სართულებზე.

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

getExceptBounds [] = []

getExceptBounds ((Flat x y z,a):xs) =

       if (x/=z)&&(x/=1) then (Flat x y z,a):getExceptBounds xs

       else getExceptBounds xs

getExceptBounds (\_:xs) = getExceptBounds xs

 გამოძახების მაგალითი:

getExceptBounds [(Flat 3 150 10,1000),(Flat 5 150 10,1000),(Flat 7 150 10,1000),(Flat 1 100 10,900)]

[(Flat 3 150 10,1000),(Flat 5 150 10,1000),(Flat 7 150 10,1000)]

it :: [(NedvObject, Int)]

1. getByType - მეორე არგუმენტად გადაეცემა სტრიქონი და ბაზიდან შესაბამის ინფორმაციას იღებს. მაგალითად,

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

getByType [] \_ = []

getByType (x:xs) t = case x of

       (Flat xx y z,a) -> if t=="Flat" then x:getByType xs t

                          else getByType xs t

       (Room xx y z zz,a) -> if t=="Room" then x:getByType xs t

                             else getByType xs t

       (House y,a) -> if t=="House" then x:getByType xs t

                      else getByType xs t

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

[(Room 4 120 9 20,1000)]

it :: [(NedvObject, Int)]