**სავარჯიშოების კრებული**

1. განსაზღვრეთ შემდეგი ფუნქციები λ ნოტაციის გამოყენებით და მიუთითეთ შესაბამისი ტიპები:
2. ფუნქცია, რომელიც აბრუნებს სიის მეორე ელემენტს.
3. ფუნქცია isPalindrome, რომლის არგუმენტია სტრიქონი და რომელიც ამოწმებს, არის თუ არა არგუმენტი პალინდრომი (იკითხება ერთნაირად მარჯვნიდან მარცხნივ და პირიქით).
4. ფუნქცია, რომელიც აწყვილებს თავის ორ არგუმენტს.
5. ფუნქცია, რომელიც გადაცემული სიისთვის აბრუნებს ლუწი ელემენტების კვადრატებს, კენტების -კუბებს.
6. ფუნქცია, რომელიც მოცემულ წყვილში ადგილებს უცვლის ელემენტებს.
7. ფუნქცია, რომელიც გადაცემული რიცხვების სიისთვის ამოწმებს თითოეული ელემენტის ლუწ-კენტობას და გვიბრუნებს ლოგიკური მნიშვნელობების სიას.
8. ფუნქცია, რომელიც აბრუნებს გაორმაგებულ არგუმენტს.
9. ფუნქცია, რომელსაც გადაეცემა ორი არგუმენტი და ითვლის მთელ ხარისხს- (ნამდვილ ხარისხს).
10. ფუნქცია twice f x, რომელიც ორჯერ ითვლის მნიშვნელობას არგუმენტზე. მაგალითად, twice succ 5=7
11. ფუნქცია, რომელსაც გადაეცემა ორი სია და დააბრუნებს მათ გაერთიანებას.
12. ფუნქცია, რომელსაც გადაეცემა ორი წყვილი და დააბრუნებს შესაბამისი ელემენტების ჯამების წყვილს. მაგალითად, არგუმენტებია: (1,2), (3,4), შედეგია: (4,6).
13. ფუნქცია, რომელსაც გადაეცემა ელემენტი და სია და დააბრუნებს ახალ სიას, სადაც პირველ (ბოლო) ელემენტად ჩასმულია პირველი არგუმენტი.

2. განსაზღვრეთ ფუნქციები:

1) ფუნქცია max3, რომელიც სამი მთელი რიცხვიდან აბრუნებს მათ შორის უდიდესს.

2) ფუნქცია min3, რომელიც სამი მთელი რიცხვიდან აბრუნებს მათ შორის უმცირესს.

3) ფუნქცია sort2, რომელიც ორი მთელი რიცხვიდან აბრუნებს წყვილს, რომელშიც პირველ ადგილას დგას ამ ორი რიცხვიდან უმცირესი, მეორეზე კი – უდიდესი.

4) ფუნქცია bothTrue :: Bool -> Bool -> Bool, რომელიც აბრუნებს True–ს, მაშინ და მხოლოდ მაშინ, როცა ორივე არგუმენტი არის True. ფუნქციის განსაზღვრისათვის არ გამოიყენოთ ლოგიკური ოპერაციები (&&, || და ა.შ.)

5) ფუნქცია solve2::Double->Double->(Bool,Double), რომელიც ორი რიცხვის მიხედვით, რომლებიც წარმოადგენენ ax + b = 0 წრფივი განტოლების კოეფიციენტებს, აბრუნებს წყვილს, რომლის პირველი ელემენტი არის True, თუ არსებობს ამონახსნი და False – წინააღმდეგ შემთხვევაში; წყვილის მეორე ელემენტი კი არის ან ფესვის მნიშვნელობა, ან 0.0.

6) ფუნქცია isParallel, რომელიც აბრუნებს True–ს, თუ ორი მონაკვეთი, რომლებიც წარმოადგენენ ფუნქციის არგუმენტებს, არის პარალელური (ან დევს ერთ წრფეზე). მაგალითად, მნიშვნელობა გამოსახულებისა isParallel (1,1) (2,2) (2,0) (4,2) არის True, ვინაიდან მონაკვეთები (1, 1) − (2, 2) და (2, 0) − (4, 2) პარალელურია.

7) ფუნქცია isIncluded, რომლის არგუმენტებია სიბრტყეზე ორი წრეწირის პარამეტრები (ცენტრის კოორდინატები და რადიუსები); ფუნქცია აბრუნებს True–ს, თუ მეორე წრეწირი მთლიანად თავსდება პირველის შიგნით.

8) ფუნქცია isRectangular, რომელიც პარამეტრად ღებულობს სიბრტყეზე სამი წერტილის კოორდინატებს და აბრუნებს True–ს, თუ მათ მიერ შედგენილი სამკუთხედი არის მართკუთხა სამკუთხედი.

9) ფუნქცია isTriangle, რომელიც განსაზღვრავს, შეიძლება თუ არა მოცემულ x, y და z სიგრძის მონაკვეთებზე აიგოს სამკუთხედი.

10) ფუნქცია isSorted, რომელიც შესასვლელზე ღებულობს სამ რიცხვს და აბრუნებს True, თუ ეს რიცხვები დალაგებულია ზრდადობით ან კლებადობით.

8. 1.იპოვეთ მნიშვნელობა (x ნამდვილია): 

2. იპოვეთ მნიშვნელობა(x ნამდვილია): 

1. ლოკალური ცვლადების გამოყენებით განსაზღვრეთ შემდეგი გამოსახულების მნიშვნელობის გამოთვლის ფუნქცია:

y= x\*u/z, სადაც х = а + b, u=b+c, z=c+a.

1. ლოკალური ცვლადების გამოყენებით განსაზღვრეთ შემდეგი გამოსახულების მნიშვნელობის გამოთვლის ფუნქცია:

f=q/w\*e, სადაც q=x-2-y, w=x\*y, e=x+y.

1. ლოკალური ცვლადების გამოყენებით განსაზღვრეთ შემდეგი გამოსახულების მნიშვნელობის გამოთვლის ფუნქცია:

x=a\*d, სადაც a=m+n\*p, d=k\*m.

1. ფუნქციის ქვემოთ მოყვანილი განმარტების მიხედვით განსაზღვრეთ, რა მნიშვნელობა მიიღება და როგორი თანმიმდევრობით:

Prelude> fun 5

Prelude> fun 1

fun n =

let funF m = if m == n then [m] else m:funF (m + 1)

in funF 1

1. ფუნქციის ქვემოთ მოყვანილი განმარტების მიხედვით განსაზღვრეთ, რა მნიშვნელობა მიიღება და როგორი თანმიმდევრობით:

Prelude> fun 5 4

Prelude> fun 4 (–7)

fun x 0 = 1

fun x n = if n > 0 then

x \* fun x (n - 1)

else error "fun: negative argument"

6 . ფუნქციის ქვემოთ მოყვანილი განმარტების მიხედვით განსაზღვრეთ, რა მნიშვნელობა მიიღება და როგორი თანმიმდევრობით:

Prelude> fun 7

Prelude> fun (–5).

fun 0 = 1

fun n | n < 0 = error "fun: negative argument"

| n >= 0 = n + fun (n - 1)