1. შეადგინეთ შემდეგი სასრული სიები (N — სიებში ელემენტების რაოდენობაა). ამისთვის გამოიყენეთ ან სიების გენერატორი ან კონსტრუქტორი ფუნქციები:

a. ნატურალური რიცხვების სია. N = 20.

b. კენტი ნატურალური რიცხვების სია. N = 20.

c. ლუწი ნატურალური რიცხვების სია. N = 20.

d. ორის ხარისხების სია. N = 25.

e. სამის ხარისხების სია. N = 25.

**ამოცანები სიის კონსტრუქტორის გამოყენებით**

1. თუ ჩამოვთვლით 10–ზე ნაკლებ ნატურალური რიცხვების სიას, რომლებიც არიან 3–ის ან 5–ის ჯერადები, მივიღებთ, რომ ასეთი ელემენტების ჯამი ტოლია 23–ის. რისი ტოლია 1000000–ზე ნაკლები 3–ის ან 5–ის ჯერადი ნატურალური რიცხვების რაოდენობა და ჯამი ? (ჩაწერეთ გამოსახულება)
2. თუ ჩამოვთვლით 10–ზე ნაკლებ ნატურალური რიცხვების სიას, რომლებიც არიან 3–ის ან 7–ის ჯერადები, მივიღებთ, რომ ასეთი ელემენტების ნამრავლი ტოლია 3\*6\*7\*9=1134–ის. რისი ტოლია 1000–ზე ნაკლები 3–ის ან 7–ის ჯერადი ნატურალური რიცხვების ნამრავლი ? (ჩაწერეთ გამოსახულება)
3. თუ ჩამოვთვლით 10–ზე ნაკლებ ნატურალური რიცხვების სიას, რომლებიც არიან 2–ის ან 7–ის ჯერადები, მივიღებთ, რომ ასეთი ელემენტების ჯამი ტოლია 27–ის. რისი ტოლია 1000–ზე ნაკლები ლუწი 7–ის ჯერადი რიცხვების რაოდენობა და მათი ჯამი.
4. რამდენით მეტია 100000–ზე ნაკლები 2–ის ან 3–ის ჯერადი ნატურალური რიცხვების ნამრავლი ამავე რიცხვების ჯამზე (ჩაწერეთ გამოსახულება) ? დათვალეთ ასეთი რიცხვების რაოდენობა.
5. (x,y,z) dadebiTi mTeli ricxvebis sameuls piTagoras sameuli ewodeba, Tu x2+y2=z2. siis konstruqtoris saSualebiT gansazRvreT funqcia: pyths :: Int –>[(Int,Int,Int)] igi gvibrunebs piTagoras yvela sameuls, romelTa komponentebi [1..n] siidan mocemul zRvrul n sidides ar aRemateba. magaliTad:

> pyths 5

[(3,4,5),(4,3,5)]

1. რა შედეგი მიიღება შემდეგი გენერატორის გამოყენებით:

ა. [(x,y) | x ← [2..3], y ← [x..4]]

ბ. [x | x ← [15..25], odd x]

გ. fs :: Int → [Int]

fs n = [x | x ← [1..n], n `mod` x == 0]

> fs 27

1. რა შედეგი მიიღება შემდეგი გენერატორის გამოყენებით:

ა. [(x,y) | x ← [2..3], y ← [x..4]]

ბ. [x | x ← [15..25], even x]

გ. cnc :: [[a]] → [a]

cnct xss = [x | xs ← xss, x ← xs]

* cnc [[1,2],[2,3],[3,4,5]]

1. იპოვეთ მოცემულ n რიცხვამდე ისეთი რიცხვები, რომელიც ორი მარტივი რიცხვის ნამრავლით (ჯამით) წარმოდგება. შედეგი წარმოადგინეთ სამეულების სახით. [(x,y,x\*y)| x<-[1..1000],y<-[1..1000], prime x,prime y]
2. შეადგინეთ ისეთი (x,y,z) სამეულების სია, სადაც თითოეული ცვლადი არის მარტივი, მნიშვნელობით 1–დან n-მდე და ჭეშმარიტია პირობა x+y==z. [(x,y,z)| x<-[1..1000],y<-[1..1000],z<-[1..1000], prime x,prime y, prime z, x+y==z]
3. იპოვეთ მოცემულ n რიცხვამდე მარტივი რიცხვების ჯამი.

sum[x|x<-[1..1000], prime x]

1. მოცემულია შემდეგი ტიპის ფუნქცია fun :: Num a => [[a]] -> a.

იპოვეთ ყველა ელემენტის ჯამი. მაგ. fun [[1,2,3],[5,4]]=15. fun xss=sum[x|xs<-xss,x<-xs]

funrec1 []=0

funrec1 (x:xs)= sum x+funrec1 xs

1. მოცემულია შემდეგი ტიპის ფუნქცია fun :: Num a => [[a]] -> [a]. იპოვეთ თითოეული ქვესიისთვის ელემენტის ჯამი. მაგ. fun [[1,2,3],[5,4]]=[6,9].

fun xs=[sum x|x<-xs]

funrec []=[]

funrec (x:xs)=(sum x):funrec xs

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

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

[x|x<-[1..1000], prime x] // 1000-მდე მარტივი რიცხვების სია.

sum[x|x<-[1..1000], prime x] --jami

76127

**ამოცანები სიების შედგენაზე**

1. შეადგინეთ იმ სამნიშნა რიცხვების სია, რომელთა ციფრთა ჯამი არის ლუწი რიცხვი

[x|x<-[100..999], even(x`mod`10 + x `div`10 `mod`10+x`div`100)]

1. შეადგინეთ n–მდე კენტი 3–ის ჯერადი რიცხვების სია.

let n=100

[x|x<-[1..n], odd x, mod x 3==0]

1. შეადგინეთ იმ ორნიშნა რიცხვების სია, რომელთა ციფრთა ჯამი არის კენტი რიცხვი.

[x|x<-[10..99], odd (x`mod`10 + x `div`10 )]

1. შეადგინეთ იმ სამნიშნა რიცხვების სია, რომელთა ციფრთა ჯამი არის 3-ის ჯერადი რიცხვი.

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

1. სამნიშნა რიცხვის ყველა ციფრი განსხვავებულია

[x\*10+y|x<-[1..9],y<-[0..9],x/=y] ორნიშნა

[x\*100+y\*10+z|x<-[1..9],y<-[0..9],z<-[0..9],x/=y&&x/=z&&y/=z ]

1. შეადგინეთ ხუთნიშნა რიცხვების სია, სადაც ყველა ციფრი ერთნაირია

[x\*10000+x\*1000+x\*100+x\*10+x|x<-[1..9]]

[11111,22222,33333,44444,55555,66666,77777,88888,99999]