Каріна Зубко 4 варіант

Лаболаторна 6:

import scala.math.pow

//Zubko

object Main extends App {

// Task 1

def geometricMean(list: List[Double]): Double = {

val n = list.length

if (n > 0) {

val product = list.foldRight(1.0)(\_ \* \_)

pow(product, 1.0 / n)

} else {

0.0

}

}

val list1 = List(2.0, 4.0, 8.0)

println(s"Task 1: ${geometricMean(list1)}")

// Task 2

def scalarProduct(listA: List[Double], listB: List[Double]): Double = {

val products = listA zip listB map { case (a, b) => a \* b }

products.foldRight(0.0)(\_ + \_)

}

val listA = List(1.0, 2.0, 3.0)

val listB = List(4.0, 5.0, 6.0)

println(s"Task 2: ${scalarProduct(listA, listB)}")

// Task 3

def countNegat(list: List[Double]): Int = {

list.count(\_ < 0)

}

val list3 = List(1.0, -2.0, 3.0, -4.0, 5.0)

println(s"Task 3: ${countNegat(list3)}")

}

