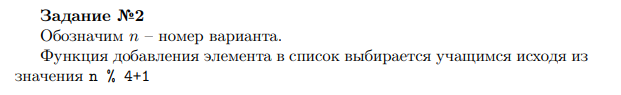


Создайте функцию, которая по данным функциям с параметром любого типа и результатами типа Int возвращает новую функцию – минимум данных (количество исходных функций – любое).

fun <T> minFunction(vararg functions: (T) -> Int): (T) -> Int {  
 return **{** input **->** functions.*map* **{ it**(input) **}**.*minOrNull*() ?: 0  
 **}**}  
  
fun main() {  
 val function1: (Double) -> Int = **{** x **->** (x \* 2).toInt() **}** val function2: (Double) -> Int = **{** x **->** (x / 2).toInt() **}** val function3: (Double) -> Int = **{** x **->** x.toInt() **}** val minFunction = *minFunction*(function1, function2, function3)  
  
 val result = minFunction(7.5)  
 *println*("Минимум: $result")  
}



Обозначим n – номер варианта. Функция добавления элемента в список выбирается учащимся исходя из значения n % 4+1

1 вариант



1. fun push (el: T): Bool вставляет элемент в начало списка;

class ListNode<T>(var value: T, var next: ListNode<T>? = null)  
  
class LinkedList<T> {  
 var head: ListNode<T>? = null  
  
 fun push(el: T) {  
 val newNode = ListNode(el)  
 newNode.next = head  
 head = newNode  
 }  
  
 fun display() {  
 var current = head  
 while (current != null) {  
 *print*("${current.value} -> ")  
 current = current.next  
 }  
 *println*("null")  
 }  
}  
  
fun main() {  
 val linkedList = LinkedList<Int>()  
  
 linkedList.push(3)  
 linkedList.push(7)  
 linkedList.push(11)  
  
 linkedList.display()  
}

сделать остальные