

ЛАБОРАТОРНАЯ РАБОТА № 7

Решение задач общего вида с использованием парадигмы функционального программирования

1.	Допустим, требуется зашифровать сообщения, состоящие из определенных символов. Например, исходный текст состоит из цифры от 0 до 9 (представленных в программе в виде символов) и других знаков и имеет длину в 15 символов. В шифровке на место цифр подставляются соответствующие им знаки, на место всех остальных символов дополнительный 11-й знак.	
2.	Дан массив из двадцати элементов. Затем находит самую длинную последовательность из нулей и выводит на экран ее длину и номер ее начала в массиве.	
3.	<p>Дана следующая информация (однако, вы можете проверить ее самостоятельно):</p> <ul style="list-style-type: none"> • 1 января 1900 года - понедельник. • В апреле, июне, сентябре и ноябре 30 дней. В феврале 28 дней, в високосный год - 29. В остальных месяцах по 31 дню. • Високосный год - любой год, делящийся нацело на 4, однако последний год века (XX00) является високосным в том и только том случае, если делится на 400. <p>Сколько воскресений выпадает на первое число месяца в двадцатом веке (с 1 января 1901 года до 31 декабря 2000 года)?</p>	
4.	Способ возведения в квадрат двузначные (и даже некоторые трехзначные) числа, оканчивающиеся на 5 заключается в следующем: для возведения в квадрат числа, оканчивающегося на 5 достаточно умножить число, полученное из исходного вычеркиванием последней пятерки на следующее по порядку число, затем остается лишь приписать «25» к получившемуся результату справа. Например, для того, чтобы возвести число 125 в квадрат достаточно 12 умножить на 13 и приписать 25, т.е. приписывая к числу $12 \cdot 13 = 156$ число 25, получаем результат 15625, т.е. $125^2 = 15625$. Напишите программу, возводящую число, оканчивающееся на 5, в квадрат.	
5.	Напишите программу, которая бы по загаданному и предложенному числам сообщала количество быков и коров. (По правилам игры «Быки и коровы»)	<p>Входные данные: 5671 7251 Вывод программы: 1 2 Входные данные: 1234 1234 Вывод программы: 4 0 Входные данные: 2034 6234</p>

		Вывод программы: 2 1
6.	Двоичное число, введенное по запросу с клавиатуры , в случае правильной записи преобразовать в шестнадцатеричное и десятичное число и результат вывести на экран.	
7.	Написать программу определения количества шестизначных " счастливых " трамвайных билетов , у которых сумма первых трех цифр совпадает с суммой трех последних .	
8.	Дан список чисел, который может содержать до 100000 чисел. Определите, сколько в нем встречается различных чисел.	Входные данные: 1 2 3 2 1 Вывод программы: 3 Входные данные: 1 2 3 4 5 6 7 8 9 10 Вывод программы: 10 Входные данные: 1 2 3 4 5 1 2 1 2 7 3 Вывод программы: 6
9.	На входе - текст. Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов. Определите, сколько различных слов содержится в этом тексте.	Входные данные: She sells sea shells on the sea shore; The shells that she sells are sea shells I'm sure. So if she sells sea shells on the sea shore, I'm sure that the shells are sea shore shells. Вывод программы: 19
10.	На вход подаются число в двоичной системе счисления. Выполнить кодирование Грея.	
11.	Разложить багаж заданного объема по ячейкам заданных размеров (размер характеризуется объемом). Выведите объем багажа, который можно разметить в ячейках.	Входные данные: 1 2 3 4 5 6 7 Вывод программы: 1 3 6 10 15 21 28 Входные данные: 1 9 5 2 Вывод программы: 1 10 15 17 Входные данные: 100000000000 Вывод программы: 100000000000

12.	<p>XOR для произвольного числа аргументов определяется следующим образом:</p> $\text{xor}(a_1 a_2 \dots, a_n) = \text{xor}(a_1 \text{ xor}(a_2 \text{ xor}(a_3 \dots \text{ xor}(a_n)) \dots))$ <p>Посчитайте XOR от n последовательностей равной длины k.</p> <p>Формат ввода</p> <p>На первой строке записано число $2 \leq n \leq 1000$ — количество последовательностей.</p> <p>На последующих n строках записаны последовательности A_1, \dots, A_n из 0 и 1, разделённых пробелами равной длины $1 \leq k \leq 1000$.</p> <p>Формат вывода</p> <p>Выведите последовательность $C = \text{xor}(A_1, \dots, A_n)$, разделяя числа последовательности пробелами.</p>	<p>Входные данные:</p> <pre>2 0 0 1 1 0 1 0 1</pre> <p>Вывод программы:</p> <pre>0 1 1 0</pre> <p>Входные данные:</p> <pre>3 0 0 0 0 1 1 1 1 0 1 0 1</pre> <p>Вывод программы:</p> <pre>1 0 1 0</pre> <p>Входные данные:</p> <pre>4 0 0 1 0 1 0 1 0 0 0 1 1 0 1 1 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 0 1 0 0 0 0 1 1 1 0 0 0 0 0</pre> <p>Вывод программы:</p> <pre>0 1 1 1 1 1 1 1 0 0</pre>
13.	<p>Выведите все простые на отрезке $[2; n]$.</p> <p>Формат ввода</p> <p>Вводится число $2 \leq n \leq 100000$.</p> <p>Формат вывода</p> <p>Выведите все простые числа из отрезка $[2, n]$ в порядке возрастания</p> <p>Примечания</p> <p>Напомним, что проверить число на то, простое ли оно можно за количество операций порядка \sqrt{N}. Также напомним, что функция <code>math.sqrt</code> работает значительно быстрее, чем $(x ** 1/2)$.</p>	<p>Входные данные:</p> <pre>8</pre> <p>Вывод программы:</p> <pre>2 3 5 7</pre> <p>Входные данные:</p> <pre>10</pre> <p>Вывод программы:</p> <pre>2 3 5 7</pre> <p>Входные данные:</p> <pre>5</pre> <p>Вывод программы:</p> <pre>2 3 5</pre>
14.	<p>Перед началом тараканьих бегов всем болельщикам было предложено сделать по две ставки на результаты бегов. Каждая ставка имеет вид "Таракан №А придет раньше, чем таракан №В". Организаторы бегов решили выяснить, могут ли тараканы прийти в таком порядке, чтобы у каждого болельщика сыграла ровно одна ставка из двух (то есть чтобы ровно одно из двух утверждений каждого болельщика оказалось верным). Считается, что никакие два таракана не могут прийти к финишу одновременно.</p> <p>Формат ввода</p> <p>В первой строке входных данных содержатся два разделенных пробелом натуральных числа: число K, не превосходящее 10, - количество тараканов и число N, не превосходящее 100, - количество болельщиков. Все тараканы пронумерованы числами от 1 до K. Каждая из следующих N строк содержит 4 натуральных</p>	<p>Входные данные:</p> <pre>3 2 2 1 2 3 1 2 3 2</pre> <p>Вывод программы:</p> <pre>3 2 1</pre> <p>Входные данные:</p> <pre>3 4 1 2 1 3 1 2 3 1 1 2 2 3 1 2 3 2</pre> <p>Вывод программы:</p> <pre>0</pre>

	<p>числа A, B, C, D, не превосходящих K, разделенных пробелами. Они соответствуют ставкам болельщика "Таракан №A придет раньше, чем таракан №B" и "Таракан №C придет раньше, чем таракан №D".</p> <p>Формат вывода</p> <p>Если завершить бега так, чтобы у каждого из болельщиков сыграла ровно одна из двух ставок, можно, то следует вывести номера тараканов в том порядке, в котором они окажутся в итоговой таблице результатов (сначала номер таракана, пришедшего первым, затем номер таракана, пришедшего вторым и т. д.) в одну строку через пробел. Если таких вариантов несколько, выведите любой из них. Если требуемого результата добиться нельзя, выведите одно число 0.</p>	
15.	<p>Во входной строке записана последовательность чисел через пробел. Для каждого числа выведите слово YES (в отдельной строке), если это число ранее встречалось в последовательности или NO, если не встречалось.</p>	<p>Входные данные: 1 2 3 2 3 4 Вывод программы: NO NO NO YES YES NO</p>
16.	<p>На Новом проспекте для разгрузки было решено пустить два новых автобусных маршрута на разных участках проспекта. Известны конечные остановки каждого из автобусов. Определите количество остановок, на которых можно пересест с одного автобуса на другой.</p> <p>Формат ввода</p> <p>Вводятся четыре числа, не превосходящие 100, задающие номера конечных остановок. Сначала для первого, потом второго автобуса (см. примеры и рисунок).</p> <p>Формат вывода</p> <p>Ваша программа должна выводить одно число — искомое количество остановок.</p>	<p>Входные данные: 3 6 4 2</p> <p>Вывод программы: 2</p> <p>Тест 2 Входные данные: 3 1 5 10</p> <p>Вывод программы: 0</p>
17.	<p>В обувном магазине продается обувь разного размера. Известно, что одну пару обуви можно надеть на другую, если она хотя бы на три размера больше. В магазин пришел покупатель. Требуется определить, какое наибольшее количество пар обуви сможет предложить ему продавец так, чтобы он смог надеть их все одновременно.</p> <p>Формат ввода</p> <p>Сначала вводится размер ноги покупателя (обувь меньшего размера он надеть не сможет), в следующей строке — размеры каждой пары обуви в магазине через пробел. Размер — натуральное число, не превосходящее 100.</p>	<p>Входные данные: 60 60 63</p> <p>Вывод программы: 2</p> <p>Тест 2 Входные данные: 26 30 35 40 41 42</p> <p>Вывод программы: 3</p>

	<p>Формат вывода</p> <p>Выведите единственное число — максимальное количество пар обуви, которое сможет надеть покупатель.</p>	<p>Тест 3</p> <p>Входные данные:</p> <p>43</p> <p>43</p> <p>Вывод программы:</p> <p>1</p>
18.	<p>Системный администратор вспомнил, что давно не делал архива пользовательских файлов. Однако, объем диска, куда он может поместить архив, может быть меньше чем суммарный объем архивируемых файлов.</p> <p>Известно, какой объем занимают файлы каждого пользователя.</p> <p>Напишите программу, которая по заданной информации о пользователях и свободному объему на архивном диске определит максимальное число пользователей, чьи данные можно поместить в архив.</p> <p>Формат ввода</p> <p>Программа получает на вход в одной строке число S – размер свободного места на диске (натуральное, не превышает 10000), и число N – количество пользователей (натуральное, не превышает 100), после этого идет N чисел - объем данных каждого пользователя (натуральное, не превышает 1000), записанных каждое в отдельной строке.</p> <p>Формат вывода</p> <p>Выведите наибольшее количество пользователей, чьи данные могут быть помещены в архив.</p>	<p>Входные данные:</p> <p>100 2</p> <p>200</p> <p>50</p> <p>Вывод программы:</p> <p>1</p> <p>Тест 2</p> <p>Входные данные:</p> <p>100 3</p> <p>50</p> <p>30</p> <p>50</p> <p>Вывод программы:</p> <p>2</p>
19.	<p>На региональном этапе Всероссийской олимпиады школьников по информатике в 2009 году предлагалась следующая задача.</p> <p>Всем известно, что со временем клавиатура изнашивается, и клавиши на ней начинают залипать. Конечно, некоторое время такую клавиатуру еще можно использовать, но для нажатий клавиш приходится использовать большую силу.</p> <p>При изготовлении клавиатуры изначально для каждой клавиши задается количество нажатий, которое она должна выдерживать. Если знать эти величины для используемой клавиатуры, то для определенной последовательности нажатых клавиш можно определить, какие клавиши в процессе их использования сломаются, а какие — нет.</p> <p>Требуется написать программу, определяющую, какие клавиши сломаются в процессе заданного варианта эксплуатации клавиатуры.</p>	<p>Входные данные:</p> <p>5</p> <p>1 50 3 4 3</p> <p>16</p> <p>1 2 3 4 5 1 3 3 4 5 5</p> <p>5 5 5 4 5</p> <p>Вывод программы:</p> <p>YES</p> <p>NO</p> <p>NO</p> <p>NO</p> <p>YES</p>

	<p>Формат ввода</p> <p>Первая строка входных данных содержит целое число n ($1 \leq n \leq 1000$) — количество клавиш на клавиатуре. Вторая строка содержит n целых чисел $c_1 \dots, c_n$, где c_i ($1 \leq c_i \leq 100000$) — количество нажатий, выдерживаемых i-ой клавишей. Третья строка содержит целое число k ($1 \leq k \leq 100000$) — общее количество нажатий клавиш, и последняя строка содержит k целых чисел p_j ($1 \leq p_j \leq n$) — последовательность нажатых клавиш.</p> <p>Формат вывода</p> <p>Программа должна вывести n строк, содержащих информацию об исправности клавиш. Если i-я клавиша сломалась, то i-ая строка должна содержать слово YES, если же клавиша работоспособна — слово NO.</p>	
20.	<p>В Государственную Думу Федерального Собрания Российской Федерации выборы производятся по партийным спискам. Каждый избиратель указывает одну партию, за которую он отдает свой голос. В Государственную Думу попадают партии, которые набрали не менее 7% от числа голосов избирателей.</p> <p>Дан список партий и список голосов избирателей. Выведите список партий, которые попадут в Государственную Думу.</p> <p>Формат ввода</p> <p>В первой строке входного файла написано слово PARTIES:. Далее идет список партий, участвующих в выборах.</p> <p>Затем идет строка, содержащая слово VOTES:. За ним идут названия партий, за которые проголосовали избиратели, по одному названию в строке. Названия могут быть только строками из первого списка.</p> <p>Формат вывода</p> <p>Программа должна вывести названия партий, получивших не менее 7% от числа голосов в том порядке, в котором они следуют в первом списке.</p>	<p>Входные данные:</p> <p>PARTIES:</p> <p>Party one</p> <p>Party two</p> <p>Party three</p> <p>VOTES:</p> <p>Party one</p> <p>Party one</p> <p>Party three</p> <p>Party one</p> <p>Party one</p> <p>Party three</p> <p>Party two</p> <p>Party one</p> <p>Party three</p> <p>Party three</p> <p>Party one</p> <p>Party one</p> <p>Party three</p> <p>Party three</p> <p>Party one</p> <p>Вывод программы:</p> <p>Party one</p> <p>Party three</p> <p>Входные данные:</p> <p>PARTIES:</p> <p>Party one</p> <p>VOTES:</p> <p>Party one</p> <p>Party one</p> <p>Party one</p> <p>Party one</p> <p>Party one</p> <p>Party one</p> <p>Party one</p> <p>Party one</p> <p>Вывод программы:</p> <p>Party one</p>
21.	<p>Формат входных данных аналогичен предыдущей задаче. Выведите</p>	<p>Входные данные:</p>

	<p>список всех партий, участвовавших в выборах, отсортировав его в порядке убывания количества голосов избирателей, а при равном количестве голосов - в лексикографическом порядке.</p>	<p>PARTIES: Party one Party two Party three VOTES: Party one Party two Party three Party two Party three Party one</p> <p>Вывод программы: Party three Party two Party one</p>
22.	<p>Дан список. Определите, является ли он монотонно возрастающим(то есть верно ли, что каждый элемент этого списка больше предыдущего).Выведите YES, если массив монотонно возрастает и NO в противном случае.Решение оформите в виде функции IsAscending(A).В данной функции должен быть один цикл while, не содержащий вложенных условий и циклов — используйте схему линейного поиска.</p>	<p>Входные данные: 1 7 9 Вывод программы: YES</p> <p>Тест 2 Входные данные: 1 9 7 Вывод программы: NO</p> <p>Тест 3 Входные данные: 2 2 2 Вывод программы: NO</p>
23.	<p>Дан список чисел. Если в нем есть два соседних элемента одного знака, выведите эти числа. Если соседних элементов одного знака нет - не выводите ничего. Если таких пар соседей несколько - выведите первую пару.</p> <p>Формат ввода</p> <p>Вводится список чисел. Все числа списка находятся на одной строке.</p>	<p>Входные данные: -1 2 3 -1 -2 Вывод программы: 2 3</p> <p>Тест 2 Входные данные: 1 -3 4 -2 1 Вывод программы:</p> <p>Тест 3 Входные данные: 1 2 -3 -4 -5 Вывод программы: 1 2</p>
24.	<p>Дан список чисел. Определите, сколько в этом списке элементов, которые больше двух своих соседей и выведите количество таких элементов.</p> <p>Формат ввода</p> <p>Вводится список чисел. Все числа списка находятся на одной строке.</p>	<p>Входные данные: 1 2 3 4 5 Вывод программы: 0</p> <p>Тест 2 Входные данные: 5 4 3 2 1 Вывод программы: 0</p>

		Тест 3 Входные данные: 1 5 1 5 1 Вывод программы: 2
25.	<p>Дан список из чисел и индекс элемента в списке k. Удалите из списка элемент с индексом k, сдвинув влево все элементы, стоящие правее элемента с индексом k.</p> <p>Программа получает на вход список, затем число k. Программа сдвигает все элементы, а после этого удаляет последний элемент списка при помощи метода pop().</p> <p>Программа должна осуществлять сдвиг непосредственно в списке, а не делать это при выводе элементов. Также нельзя использовать дополнительный список.</p> <p>Формат ввода</p> <p>Вводится список чисел. Все числа списка находятся на одной строке. В следующей строке вводится одно целое число.</p>	Входные данные: 7 6 5 4 3 2 1 2 Вывод программы: 7 6 4 3 2 1 Тест 2 Входные данные: 4 6 2 4 3 5 12 24 3 5 4 Вывод программы: 4 6 2 4 5 12 24 3 5 Тест 3 Входные данные: 4 6 2 4 3 5 12 24 3 5 0 Вывод программы: 6 2 4 3 5 12 24 3 5
26.	<p>Напишите программу, которая находит в массиве элемент, самый близкий по величине к данному числу.</p> <p>Формат ввода</p> <p>В первой строке задается одно натуральное число N, не превосходящее 1000 – размер массива. Во второй строке содержатся N чисел – элементы массива (целые числа, не превосходящие по модулю 1000). В третьей строке вводится одно целое число x, не превосходящее по модулю 1000.</p>	Входные данные: 5 1 2 3 4 5 6 Вывод программы: 5 Тест 2 Входные данные: 5 5 4 3 2 1 3 Вывод программы: 3
27.	<p>Петя перешёл в другую школу. На уроке физкультуры ему понадобилось определить своё место в строю. Помогите ему это сделать и вычислите средний рост одноклассников, которые выше.</p> <p>Формат ввода</p> <p>Программа получает на вход невозрастающую последовательность натуральных чисел, означающих рост каждого человека в строю. После этого вводится число X – рост Пети. Все числа во входных данных натуральные и не превышают 200.</p>	Входные данные: 165 163 160 160 157 157 155 154 162 Вывод программы: 3 Тест 2 Входные данные: 165 163 160 160 157 157 155 154 160 Выв
28.	<p>Дан список, заполненный произвольными целыми числами. Найдите в этом списке два числа, произведение которых максимально.</p>	Входные данные: 4 3 5 2 5

	<p>Выведите эти числа в порядке неубывания.</p>	<p>Вывод программы: 5 5</p> <p>Тест 2 Входные данные: -4 3 -5 2 5</p> <p>Вывод программы: -5 -4</p>
29.	<p>Дан список. Выведите те его элементы, которые встречаются в списке только один раз. Элементы нужно выводить в том порядке, в котором они встречаются в списке.</p> <p>Формат ввода</p> <p>Вводится список чисел. Все числа списка находятся на одной строке.</p>	<p>Входные данные: 1 2 2 3 3 3</p> <p>Вывод программы: 1</p> <p>Тест 2 Входные данные: 4 3 5 2 5 1 3 5</p> <p>Вывод программы: 4 2 1</p>
30.	<p>N кеглей выставили в один ряд, занумеровав их слева направо числами от 1 до N. Затем по этому ряду бросили K шаров, при этом i-й шар сбил все кегли с номерами от l_i до r_i включительно. Определите, какие кегли остались стоять на месте.</p> <p>Формат ввода</p> <p>Программа получает на вход количество кеглей N и количество бросков K. Далее идет K пар чисел l_i, r_i, при этом $1 \leq l_i \leq r_i \leq N \leq 100$.</p> <p>Формат вывода</p> <p>Программа должна вывести последовательность из N символов, где j-й символ — "I", если j-я кегля осталась стоять, или ".", если j-я кегля была сбита.</p>	<p>Входные данные: 10 3 8 10 2 5 3 6</p> <p>Вывод программы: I.....I...</p> <p>Тест 2 Входные данные: 5 2 1 2 4 4</p> <p>Вывод программы: ..I.I</p> <p>Тест 3 Входные данные: 10 3 3 5 4 6 10 10</p> <p>Вывод программы: II.....III.</p>
31.	<p>Дан список. Не изменяя его и не используя дополнительные списки, определите, какое число в этом списке встречается чаще всего. Если таких чисел несколько, выведите любое из них.</p> <p>Формат ввода</p> <p>Вводится список чисел. Все числа списка находятся на одной строке.</p>	<p>Входные данные: 1 2 3 2 3 3</p> <p>Вывод программы: 3</p> <p>Тест 2 Входные данные: 1 2 3 4 5 6 7 8 9 1</p> <p>Вывод программы: 1</p>

		Тест 3 Входные данные: 1 1 2 2 3 3 4 4 5 5 5 Вывод программы: 5
32.	На столе лежат n монеток. Некоторые из них лежат вверх решкой, а некоторые – гербом. Определите минимальное число монеток, которые нужно перевернуть, чтобы все монетки были повернуты вверх одной и той же стороной.	
33.	Однажды компьютерная мышка подумала, что стоит взять про запас еще один коврик. Чтобы никто не заметил запасного коврика, мышка решила его спрятать под свой прямоугольный коврик. Пробравшись ночью на склад, мышка обнаружила там только круглые коврики. Удастся ли мышке спрятать круглый коврик под прямоугольным ковриком?	