Лабораторная работа №1

Создание LINQ-запросов в C#. Поэлементные операции, теоретико-  
множественные операции

1. Дана целочисленная последовательность, содержащая как положительные,

так и отрицательные числа. Вывести ее первый положительный элемент и последний отрицательный элемент.

1. Даны цифра *D* (однозначное целое число) и целочисленная последовательность *A*. Вывести первый положительный элемент последовательности *A*, оканчивающийся цифрой *D*. Если требуемых элементов в последовательности *A* нет, то вывести 0.
2. Даны целое число *L* (> 0) и строковая последовательность *A*. Вывести последнюю строку из *A*, начинающуюся с цифры и имеющую длину *L*. Если требуемых строк в последовательности *A* нет, то вывести строку «Not found».

**Указание.** Для обработки ситуации, связанной с отсутствием требуемых строк, использовать операцию ??.

1. Даны символ *С* и строковая последовательность *A*. Если *A* содержит

единственный элемент, оканчивающийся символом *C*, то вывести этот элемент; если требуемых строк в *A* нет, то вывести пустую строку; если требуемых строк больше одной, то вывести строку «Error».

1. Даны символ *С* и строковая последовательность *A*. Найти количество элементов *A*, которые содержат более одного символа и при этом начинаются и оканчиваются символом *C*.
2. Дана целочисленная последовательность. Найти количество ее

положительных двузначных элементов, а также их среднее арифметическое (как вещественное число). Если требуемые элементы отсутствуют, то дважды вывести 0 (первый раз как целое, второй - как вещественное).

1. Даны целое число *L* (> 0) и строковая последовательность *A*. Строки

последовательности *A* содержат только заглавные буквы латинского алфавита. Среди всех строк из *A*, имеющих длину *L*, найти наибольшую (в смысле лексикографического порядка). Вывести эту строку или пустую строку, если последовательность не содержит строк длины *L*.

1. Дана строковая последовательность. Найти сумму длин всех строк, входящих в данную последовательность.
2. Даны целое число *D* и целочисленная последовательность *A*. Начиная с первого элемента *A*, большего *D*, извлечь из *A* все нечетные положительные числа, поменяв порядок извлеченных чисел на обратный.
3. Дано целое число *K* (> 0) и строковая последовательность *A*. Из элементов *A*, предшествующих элементу с порядковым номером *K*, извлечь те строки, которые имеют нечетную длину и начинаются с заглавной латинской буквы, изменив порядок следования извлеченных строк на обратный.
4. Даны целые числа *D* и *K* (*K* > 0) и целочисленная последовательность *A*. Найти теоретико-множественное объединение двух фрагментов *A*: первый содержит все элементы до первого элемента, большего *D* (не включая его), а второй - все элементы, начиная с элемента с порядковым номером *K*. Полученную последовательность (не содержащую одинаковых элементов) отсортировать по убыванию.
5. Даны целое число *K* (> 0) и целочисленная последовательность *A*. Найти теоретико-множественную разность двух фрагментов *A*: первый содержит все четные числа, а второй - все числа с порядковыми номерами, большими *K*. В полученной последовательности (не содержащей одинаковых элементов) поменять порядок элементов на обратный.
6. Даны целое число *K* (> 0) и последовательность непустых строк *A*. Строки последовательности содержат только цифры и заглавные буквы латинского алфавита. Найти теоретико-множественное пересечение двух фрагментов *A*: первый содержит 3*K* начальных элементов, а второй - все элементы, расположенные после последнего элемента, оканчивающегося цифрой. Полученную последовательность (не содержащую одинаковых элементов) отсортировать по возрастанию длин строк, а строки одинаковой длины - в лексикографическом порядке по возрастанию.
7. Дана строковая последовательность *A*. Строки последовательности содержат только заглавные буквы латинского алфавита. Получить новую последовательность строк, элементы которой определяются по соответствующим элементам *A* следующим образом: пустые строки в новую последовательность не включаются, а к непустым приписывается порядковый номер данной строки в исходной последовательности (например, если пятый элемент *A* имеет вид «ABC», то в полученной последовательности он будет иметь вид «ABC5»). При нумерации должны учитываться и пустые строки последовательности *A*. Отсортировать полученную последовательность в лексикографическом порядке по возрастанию.