

Лабораторная 7. LINQ

Сдать до 26.10

Основные задания

Задание 1 (1 балл)

Для выборки элементов (предполагая, что у каждого элемента есть имя Name) произвести конкатенацию имен всех элементов, кроме первых трех, в одну строку, разделенных заданным параметром (символом) `delimiter`.

Задание 2 (2 балла)

Найти все элементы в последовательности/выборке, длина имени (количество символов) у которых больше, чем позиция, которую они занимают в последовательности/выборке.

Задание 3 (2 балла)

Для заданного предложения сгруппировать слова одинаковой длины, отсортировать группы по убыванию количества элементов в каждой группе, вывести информацию по каждой группе: длина (количество букв в словах группы), количество элементов. Знаки препинания не учитывать.

Пример

Для предложения

"Это что же получается: ходишь, ходишь в школу, а потом бац - вторая смена"

Группа 2. Длина 6. Количество 3

ходишь

ходишь

вторая

Группа 3. Длина 5. Количество 3

школу

потом

смена

Группа 4. Длина 3. Количество 3

это

что

бац

Группа 6. Длина 1. Количество 2

в

а

Группа 1. Длина 10. Количество 1

получается

Группа 5. Длина 2. Количество 1

же

Задание 4 (2 балла)

Пусть есть англо-русский словарь. Есть некоторый текст на английском языке (представлен в виде последовательности слов). Необходимо сверстать из этих предложений книгу на русском языке

для плохо выглядящих так, что на одной странице книги помещается не более N слов и при этом каждое слово напечатано в верхнем регистре. Перевод необходимо осуществлять пословно без учета грамматики. Считается, что каждое слово имеет перевод в словаре.

Например, при N=3 входной текст:

«This dog eats too much vegetables after lunch» будет сверстан по страницам в книгу так:

ЭТА СОБАКА ЕСТ // 1 страница

СЛИШКОМ МНОГО ОВОЩЕЙ // 2 страница

ПОСЛЕ ОБЕДА // 3 страница

Реализуйте создание такой книги по входному тексту (строке) с использованием LINQ.

Задание 5. Блоки слов (3 балла)

Напишите функцию, которая разделяет фразу на сегменты (bucket) слов, каждый из которых содержит n или меньше символов. Включайте в каждый сегмент только полные слова.

Примеры

bucketize («она продает морские раковины у моря», 16) → [«она продает», «морские раковины», «у моря»]

bucketize («мышь прыгнула через сыр», 8) → «мышь», «прыгнула», «через», «сыр»]

bucketize («волшебная пыль покрыла воздух», 15) → [«волшебная пыль», «покрыла воздух», «воздух»]

bucketize («a b c d e », 2) → [«a», «b», «c», «d», «e»]

Примечания

Пробелы считаются одним символом.

Обрежьте начальные и конечные пробелы для каждого сегмента слова.

Если сегменты слишком малы, чтобы вместить одно слово, верните пустой массив: []

Конечная цель состоит не в том, чтобы вернуть только слова, длина которых равна (или меньше) заданному n, а в том, чтобы вернуть всю заданную фразу с разделением на столбцы (если это возможно).

Дополнительные задания

Задание 6. Макси и Минни (3 балла)

Макси - это наибольшее число, которое можно получить, поменяв местами две цифры, Минни - наименьшее. Напишите функцию, которая принимает число и возвращает массив длиной из двух элементов: макси и минни. Нули в начале не допускаются. Допускается не делать перестановки. Все входные данные - положительные целые числа типа ulong.

Примеры

maxmin(12340) → new long[] { 42310, 10342 }

maxmin(98761) → new long[] { 98761, 18769 }

maxmin(9000) → new long[] { 9000, 9000 }

maxmin(11321) → new long[] { 31121, 11123 }

Задание 7. Задача трех сумм (4 балла)

Напишите функцию, которая возвращает все наборы из трех элементов, сумма которых равна 0.

Примеры

ThreeSum(new int[] { 0, 1, -1, -1, 2 }) → { { -1, -1, 2 }, { 0, 1, -1 } }

ThreeSum(new int[] { 0, 0, 0, 5, -5 }) → { { 0, 0, 0 }, { 0, 5, -5 } }

ThreeSum(new int[] { 1, 2, 3 }) → { }

ThreeSum(new int[1]) → { }

Примечания

Исходный массив может содержать повторяющиеся числа. Можно вставлять нулевые элементы, которых нет в исходном массиве

Подмассивы следует упорядочивать по первому элементу подмассивов по возрастанию.

Элементы внутри подмассивов должны идти в том же порядке, что и в исходном массиве.