ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОМСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Информатика и вычислительная техника»

Отчёт по лабораторной работе № 2

по дисциплине

«Проектирование и тестирование программного обеспечения»

Выполнил:

Студент гр. ПИН-222

Лубенский И. В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подп., дата)

Проверил:

Старший преподаватель каф. ИВТ

Блохин А. В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подп., дата)

Омск, 2024

**ОПИСАНИЕ ЗАДАЧИ “ДУБЛЕТЫ”**

Дублетом называются два слова, которые отличаются ровно в одной букве (например, «booster» и «rooster», или «rooster» и «roaster», или «roaster» и «roasted»).

Задается словарь длиной не более 25 143 слов, состоящих из строчных букв, причем длина каждого слова не превышает 16 букв. Далее вам задается некоторое число пар слов. Для каждой пары слов найдите последовательность слов, имеющую наименьшую длину, причем первое слово последовательности должно совпадать с первым словом из заданной пары, а последнее слово последовательности — со вторым словом из пары. Каждая пара соседних слов последовательности должна быть дублетом. Например, если вам задана пара «booster» и «roasted», возможным решением является «booster», «rooster», «roaster», «roasted», при условии, что все эти слова присутствуют в словаре.

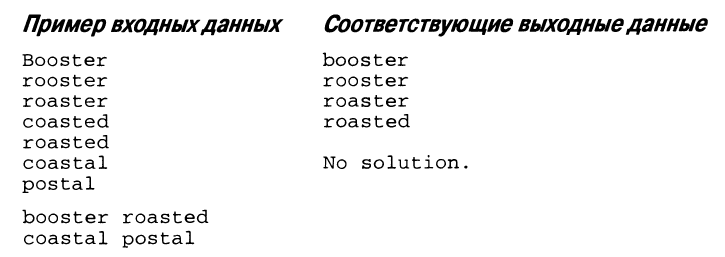
**Входные данные**

Файл входных данных содержит словарь, за которым следует некоторое число пар слов. Словарь состоит из произвольного числа слов, по одному на строку, и завершается пустой строкой. Далее идут пары слов; каждая строка содержит пару слов, разделенных пробелом.

**Выходные данные**

Для каждой введенной пары напечатайте набор строк, начинающийся с первого слова и заканчивающийся последним. Каждая пара соседних строк должна быть дублетом.

Если существует несколько вариантов минимальных решений, то подойдет любое. Если решения не существует, выведите строку "No solution". Между блоками должна быть пустая строка.



**АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ “ДУБЛЕТЫ”**

Схема алгоритма решения поставленной задачи показана на рисунке 1.

Рисунок 1 — Схема составленного алгоритма

Исходный код программы показан на рисунке 2.

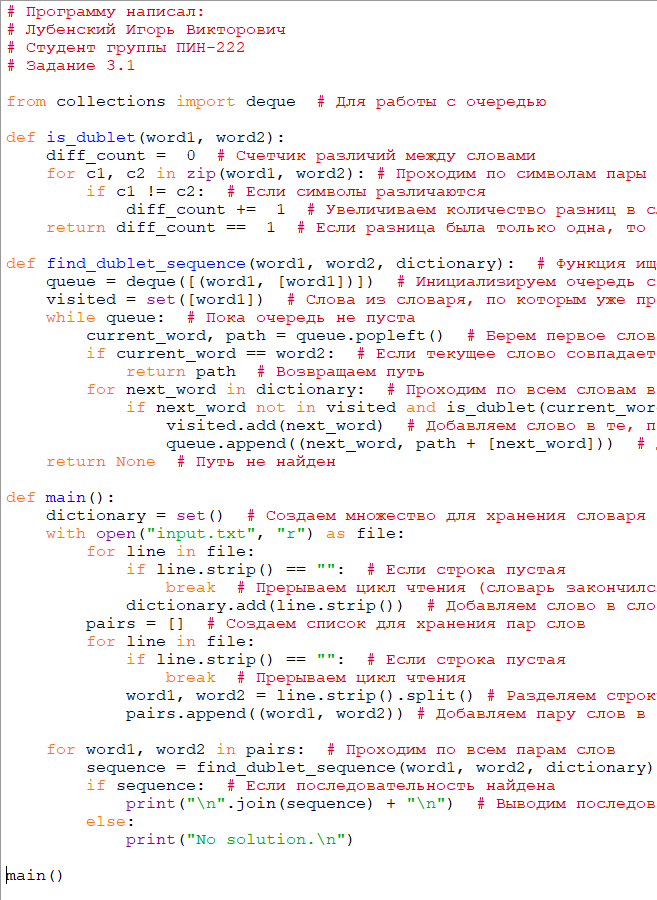


Рисунок 2 — Исходный код для решения задачи

В качестве языка программирования для решения задачи был выбран язык программирования Python.

Программа пытается открыть файл с указанным названием. Если это не удаётся сделать, то выводится ошибка о том, что файл не найден. Если файл удалось открыть, то входные данные из него считываются построчно. Из каждой строки убираются лишние пробелы, данные в каждой строке разделяются пробелами.

После считывания очередной строки программа пытается привести входные данные к типу данных int, так как по условию задачи значения i и j должны быть целыми. Кроме того, входные данные также проверяются на соответствие диапазону от 0 до 1 000 000, указанному в условии задачи. Если данные из файла не удалось привести к целым числам или они не входят в заданный диапазон, то программа выведет соответствующие ошибки.

Если входные данные в конкретной строке были валидными, то начинается работа алгоритма 3n + 1. На каждой итерации происходит подсчёт длины последовательности, которая по умолчанию равна 1.

Для каждой строки выводится число i, число j, а также длина последовательности, получаемой при отработке алгоритма от i до j. Если входные данные в конкретной строке были невалидными, то для данной строки будет выведена соответствующая ошибка и программа приступит к обработке следующей строки файла.

**ТЕСТОВЫЕ ДАННЫЕ И РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

На рисунке 3 показаны входные данные, находящиеся в файле.

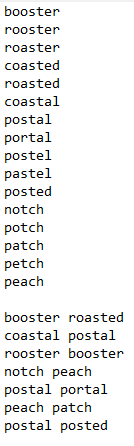


Рисунок 3 — Входные данные

Некоторые входные данные были взяты из методических указаний. Словарь был пополнен подходящими и не походящими словами и последовательностями для других тестов.

Результат обработки указанных входных данных показан на рисунке 4.

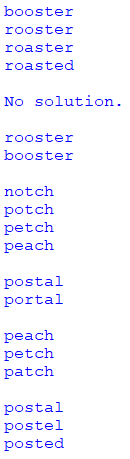


Рисунок 4 — Результат работы программы

Результаты обработки первых двух тестов сходятся в методическими указаниями. Программа верно отделяет словарь от слов, для которых необходимо искать последовательности. В случае возникновения во входных данных повторно встреченной пустой строки, программа прекратит выполнение. Сравнение программного решения с решением вручную показало правильность работы программы.

**ОПИСАНИЕ ЗАДАЧИ “СОРТИРОВКА ШЕЛЛА”**

Король Йертл хочет перегруппировать свой трон из черепах так, чтобы его самые знатные дворяне и ближайшие советники оказались ближе к вершине. Для изменения порядка черепах в груде доступна лишь одна операция: черепаха может уползти со своего места в груде и заползти по остальным черепахам наверх.

Вам задается начальный порядок груды черепах и желаемый порядок той же груды, ваша задача состоит в том, чтобы найти минимальную последовательность операций, которые преобразуют заданную груду в желаемую.

**Входные данные**

Первая строка входных данных состоит из одного целого числа К, задающего количество тестовых блоков. Каждый тестовый блок состоит из целого числа W, задающего число черепах в груде. Следующие n строк содержат начальный порядок груды черепах. Каждая строка содержит имя черепахи, начиная с черепахи, находящейся на верху груды, и заканчивая черепахой, находящейся в самом низу. Имя каждой черепахи уникально, причем каждое имя состоит не более чем из 80 символов, которые выбираются из символьного множества, состоящего из буквенно-цифровых символов, символа пробела и точки («.»).

Следующие n строк входных данных задают желаемый порядок груды, снова перечисляя имена черепах сверху вниз. Каждый тестовый блок состоит ровно из 2n + 1 строк. Число черепах (n) не превышает 200.

**Выходные данные**

Для каждого тестового блока выходные данные состоят из последовательности имен черепах - по одному на строку, означающих порядок, в котором черепахи должны уползать со своего места и заползать наверх. Эта последовательность операций должна преобразовывать начальный порядок в желаемый и иметь минимальную длину. Если подходят несколько различных решений, вы можете привести любое.

После каждого тестового блока должна быть пустая строка.

**Пример входных данных**

2

3

Yertle

Duke of Earl

Sir Lancelot

Duke of Earl

Yertle

Sir Lancelot

9

Yertle

Duke of Earl

Sir Lancelot

Elizabeth Windsor

Michael Eisner

Richard M. Nixon

Mr. Rogers

Ford Perfect

Mack

Yertle

Richard M. Nixon

Sir Lancelot

Duke of Earl

Elizabeth Windsor

Michael Eisner

Mr. Rogers

Ford Perfect

Mack

**Соответствующие выходные данные**

Duke of Earl

Sir Lancelot Richard M. Nixon Yertle

**АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ “СОРТИРОВКА ШЕЛЛА”**

Схема алгоритма решения поставленной задачи показана на рисунке 5.

Рисунок 5 — Схема алгоритма решения задачи

Исходный код программы показан на рисунке 6.

Рисунок 6 — Исходный код решения задачи

В программе создана специальная функция для проверки, является ли последовательность из n значений Jolly Jumper.

Если во входных данных в качестве n было отрицательное число, то функция сразу сообщает, что последовательность Not Jolly, потому что в данных сказано, что задана последовательность из отрицательного количества чисел, что невозможно.

В функции создаётся массив для хранения абсолютных значений разностей последовательных значений в наборе. Если абсолютная разность каких-либо значений выходит больше, чем n -1 или получается равной 0, или получается число, которое уже было среди абсолютных разностей, то программа говорит, что данная последовательность Not Jolly. Иначе полученное значение абсолютной разности добавляется в последовательность, и обработка продолжается. Если условия соблюдены, то функция вернёт строку “Jolly” для заданного n и последовательности чисел. Эта функция используется внутри основной части кода.

Программа открывает файл с входными данными и считывает его построчно. Данные в строках разделяются по пробелам. Первым элементом в строке является n, оно приводится к числу. Остальные элементы являются последовательностью, они также приводятся к числу. Если какие-либо данные не удалось привести к числовым, то программа выводит соответствующую ошибку. Если данные были успешно приведены к необходимому типу данных, то они передаются в функцию, описанную выше, и обрабатываются ей.

**ТЕСТОВЫЕ ДАННЫЕ И РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

На рисунке 7 показаны входные данные, находящиеся в файле.

Рисунок 7 — Входные данные для задачи

Первые две строчки входных данных были скопированы из методических указаний. Остальные данные являются либо валидными, либо специально неправильными, либо содержат текстовые данные или пустые данные.

Результат обработки входных данных показаны на рисунке 8.

Рисунок 8 — Результат работы программы

В программе верно построен основной алгоритм определения Jolly и Not Jolly последовательностей. Это видно из того, что выходные данные программы сходятся с примером из методических указаний.

В строке 12 входных данных в качестве значения n специально был указан символ “b”, который невозможно интерпретировать как целое число, о чём программа успешно сообщила и перешла к обработке следующих входных данных.

Строка 13 была оставлена пустой, что также было верно обработано программой и была выведена соответствующая ошибка.