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.

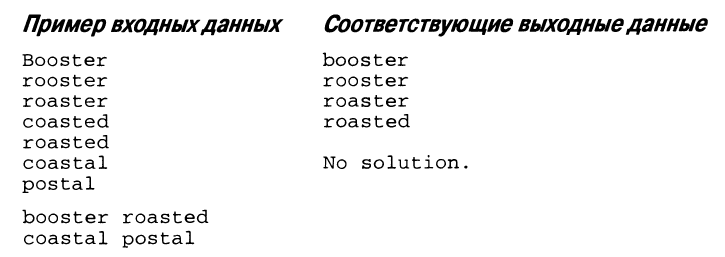
**Входные данные**

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.

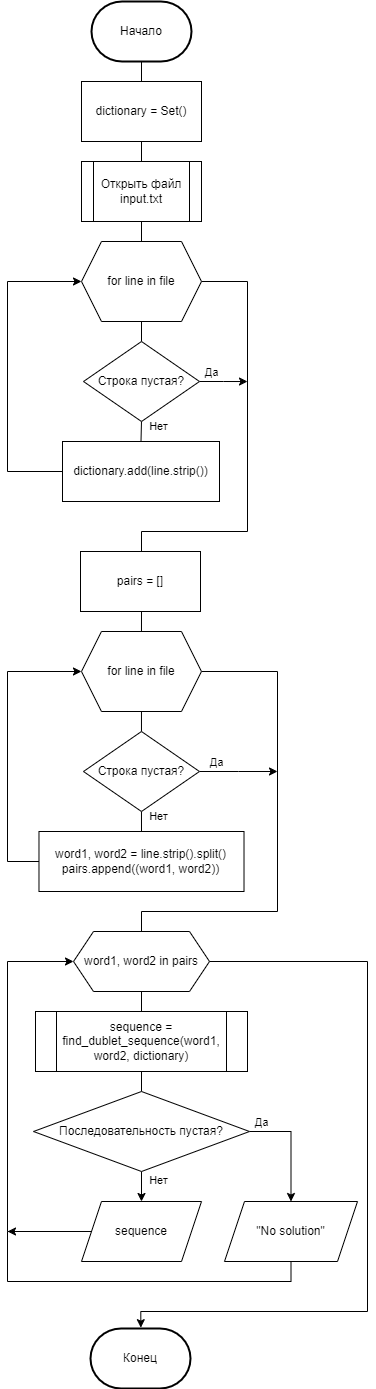


Рисунок 1 — Схема составленного алгоритма

Схемы алгоритмов функций для поиска последовательности дублетов и определения, являются ли два слова дублетами, использованных в решении задачи, показаны на рисунке 1.2.

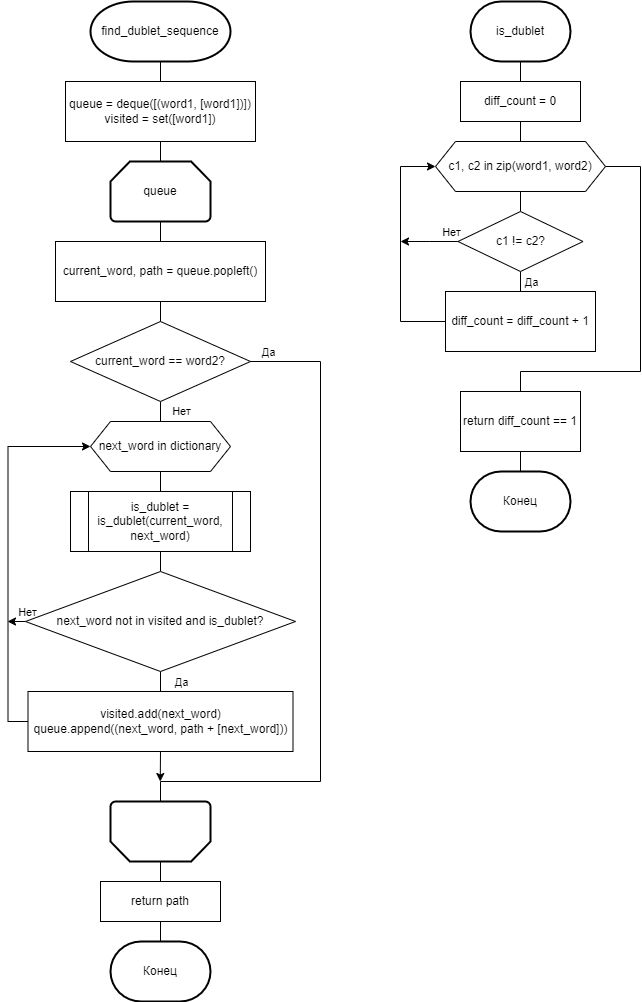


Рисунок 1.2 — Функции для работы с дублетами

Исходный код программы показан на рисунке 2.

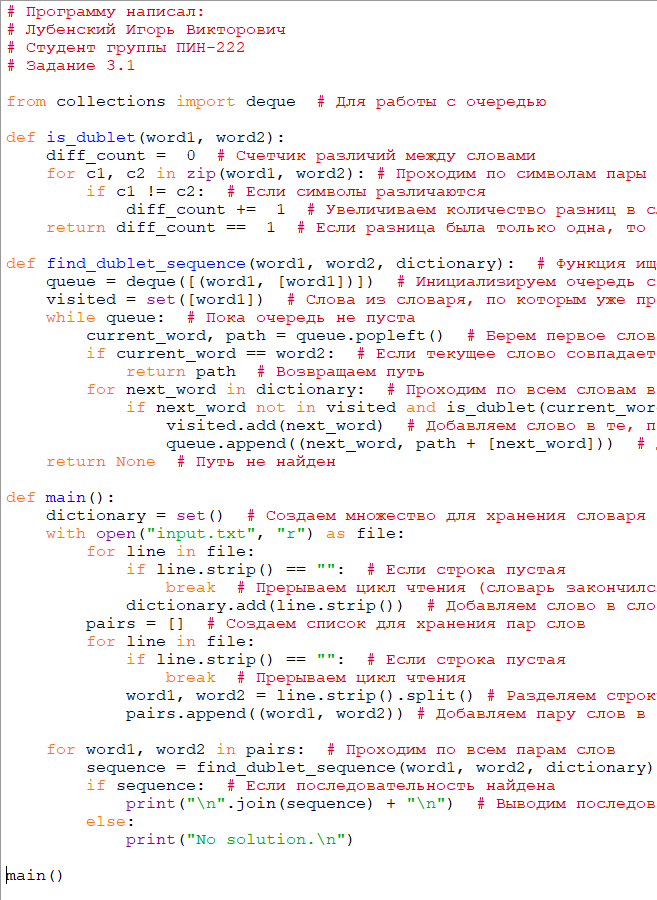


Рисунок 2 — Исходный код для решения задачи

В программе заведено две специальные функции для работы с дублетами.

Первая из них позволяет определить, являются ли два слова дублетами. Для этого она одновременно проходит посимвольно по каждому из слов и сравнивает их буквы. Если буквы отличаются, то происходит подсчёт отличий между словами. Если число отличий равно единице, то пара слов является дублетом.

Вторая функция ищет последовательность дублетов от одного слова ко второму. Для этого используется простая очередь, которая и будет представлять собой последовательность от одного слова к другому через дублеты, при этом очередь начинается со слова, от которого происходит поиск последовательности. Функция проходит по всем словам в заданном во входном файле словаре. Если конкретное слово ещё не встречалось и при этом является дублетом для данного, то оно добавляется в список посещённых слов и в очередь. Таким образом путь обновляется. Если данное слово в определённый момент совпадёт со словом, до которого рассчитывался путь, то можно сказать, что путь от одного к другому найден и программа возвращает этот путь. В противном случае, можно сказать, что пути через дублеты от первого слова до второго нет во входном словаре.

В основной функции программы создаётся множество для хранения словаря из входного файла. Это необходимо для того, чтобы проигнорировать слова, которые могли быть продублированы в словаре. Входной файл открывается для чтения, из него считывается словарь, а также тестовые данные, которые разбиваются на пары слов.

Главная функция проходит по парам слов из тестов и, используя функцию поиска последовательности дублетов, выводит найденную последовательность на экран. В противном случае программа выводит “No solution” для внесённой тестовой пары.

**ТЕСТОВЫЕ ДАННЫЕ И РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

На рисунке 3 показаны входные данные, находящиеся в файле.

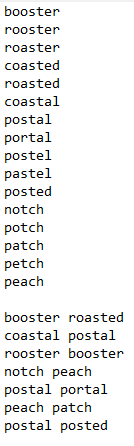


Рисунок 3 — Входные данные

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.

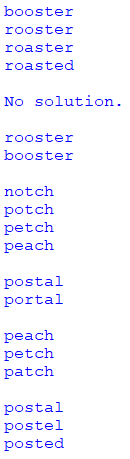


Рисунок 4 — Результат работы программы

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.

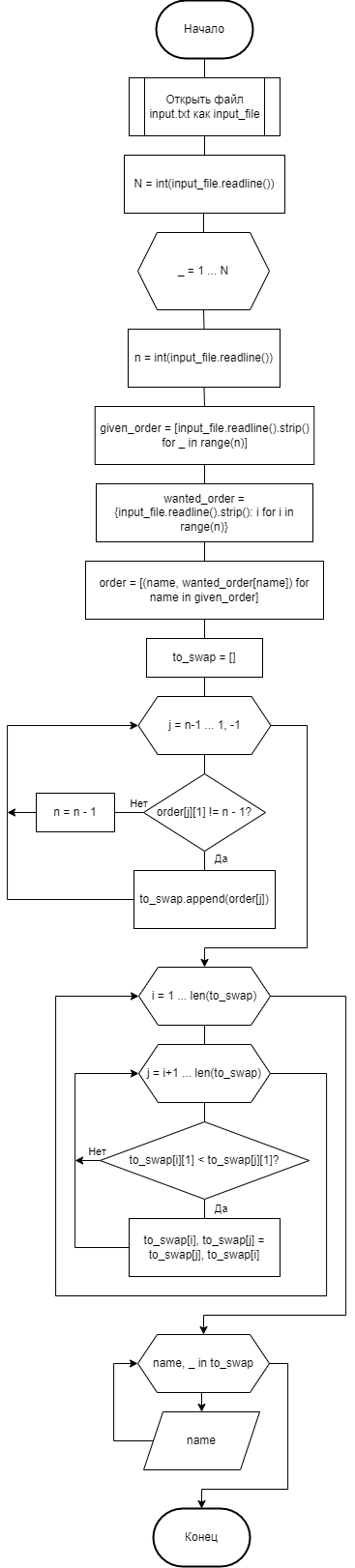


Рисунок 5 — Схема алгоритма решения задачи

Исходный код программы показан на рисунке 6.

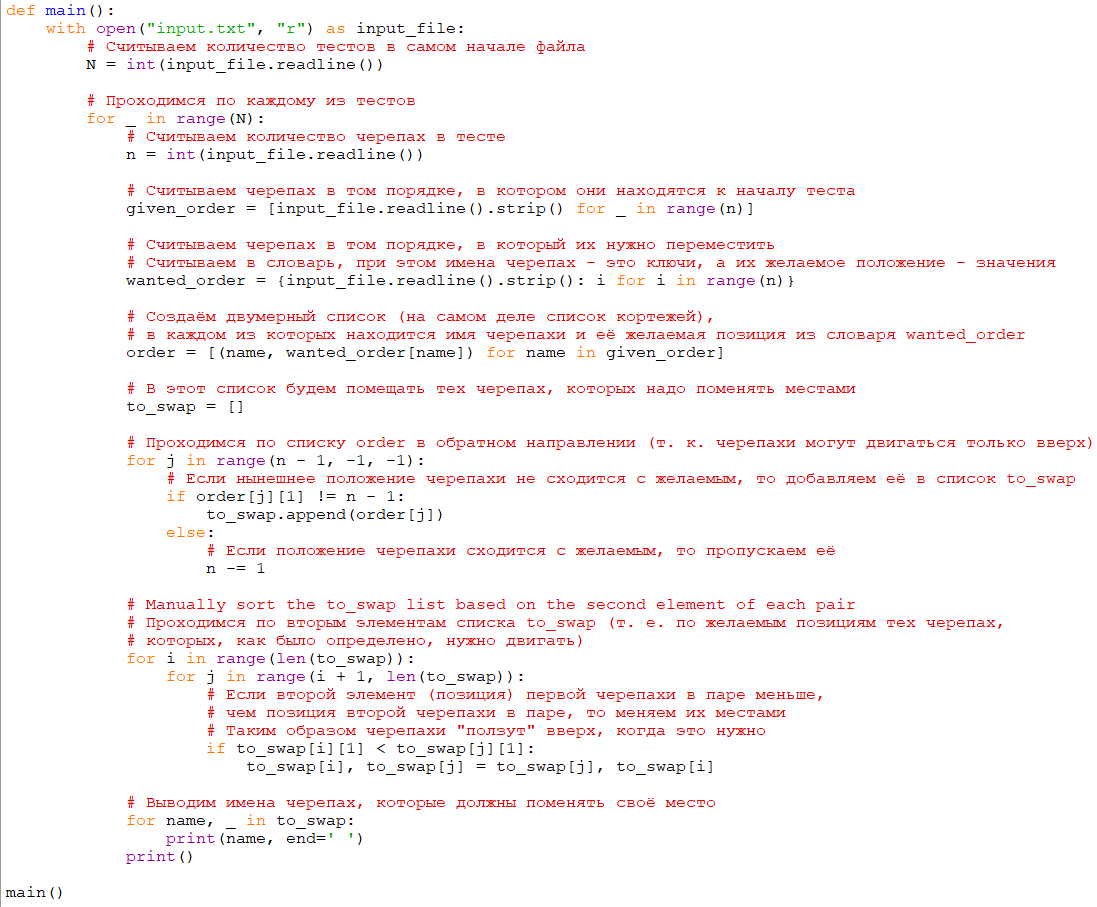


Рисунок 6 — Исходный код решения задачи

Программа открывает входной файл и считывает количество тестов, находящихся в файле из первой строки. После этого она проходит по каждому из тестов.

Для каждого из тестов в файле программа считывает количество черепах, а также порядок, в котором они находятся к началу теста и порядок, в котором они должны находиться после работы программы.

В программе создаётся специальный список, в который помещаются черепахи, которые находятся не на своём месте и должны быть передвинуты. При этом список черепах просматривается снизу вверх, так как по условию задачи черепахи могут переползать только вверх.

После определения черепах, которым необходимо передвинуться для желаемой последовательности, программа проходит по парам рядом стоящих черепах и сравнивает их нынешние позиции с желаемыми. Если данная черепаха находится ниже, чем должна, то она меняется местами со второй. Таким образом, черепахи занимают места, указанные в желаемой последовательности во входных данных.

После этого программа выводит имена черепах, которым было необходимо сменить своё место, чтобы занять желаемую позицию.

**ТЕСТОВЫЕ ДАННЫЕ И РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

На рисунке 7 показаны входные данные, находящиеся в файле.

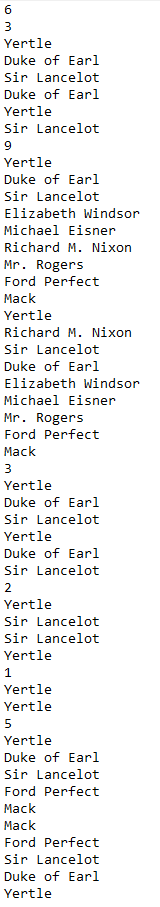


Рисунок 7 — Входные данные для задачи

Первые два теста были скопированы из методических указаний к заданию и были правильно обработаны программой. Следующие четыре теста проверяют несколько крайних случаев в задаче: когда в куче только одна черепаха, несколько черепах, но они уже находятся в желаемых позициях, а также когда в куче несколько черепах и всем им нужно переползти выше.

Результат обработки входных данных показаны на рисунке 8.

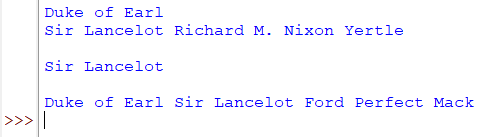


Рисунок 8 — Результат работы программы

Во всех случаях программа верно определяет оптимальные перемещения черепах, в том числе выводит пустые строки в том случае, когда ни одного перемещения с заданной кучей производить не нужно.