1. **Цель работы:**

Изучение команд адресной арифметики.

1. **Постановка задачи:**

В символьной строке удалить все слова, содержащие две подряд идущие гласные буквы.

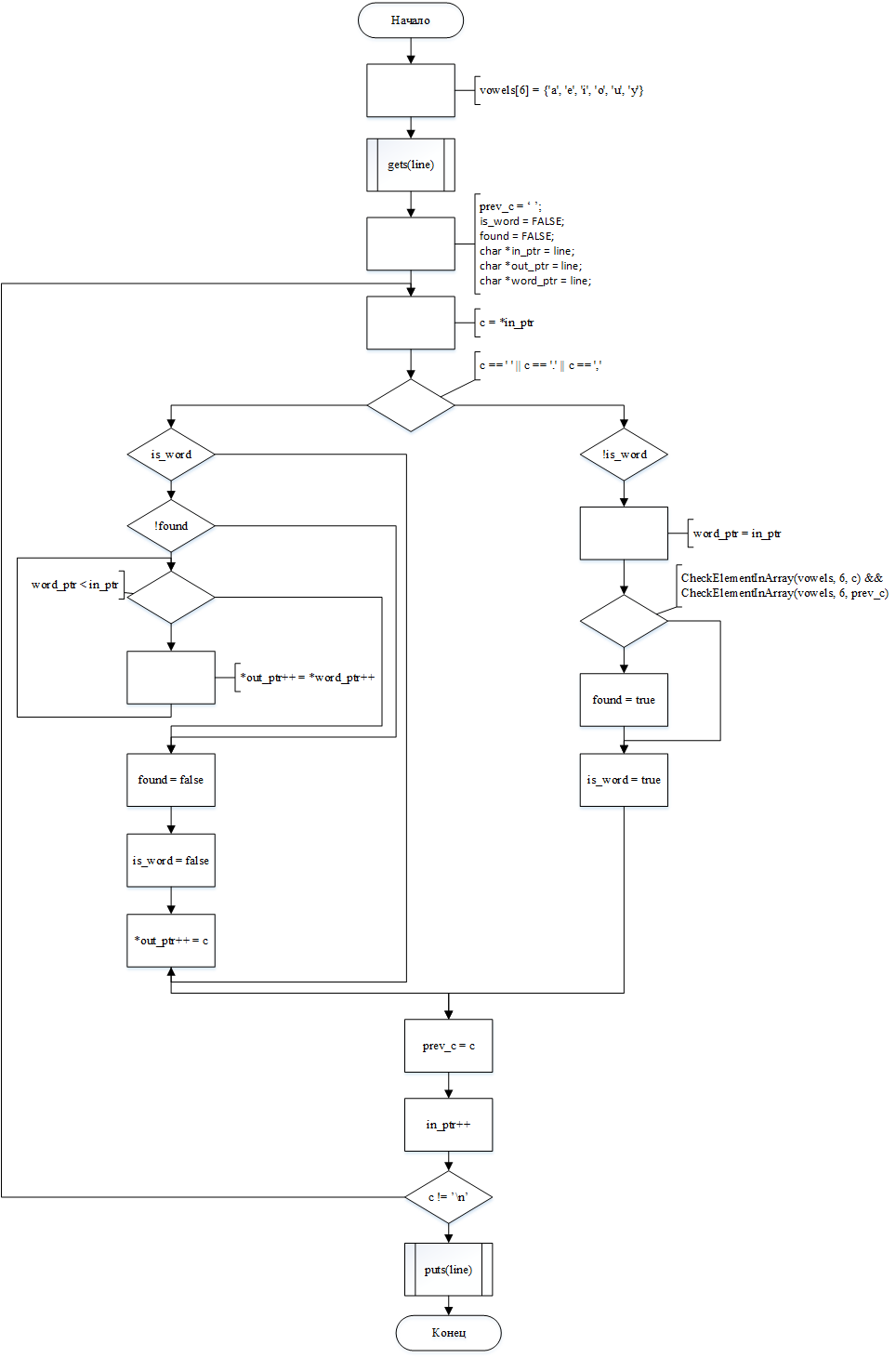
1. **Формализация:**
2. Исходная строка вводится с клавиатуры.
3. Символьная переменная с хранит в себе текущий символ, prev\_c – предыдущий. Переменные is\_word и found – это флаги, которые отвечают за признаки слова и удаления, соответственно. in\_ptr – указатель на входную строку, out\_ptr – на выходную, word\_ptr – на начало слова.
4. В цикле do-while перебираются все символы строки, пока не встретится символ перевода '\n'. В каждой итерации цикла происходит перевод указателя входящей строки in\_ptr на следующий элемент.
5. Если встречается символ-разделитель, то проверяются флаги is\_word и found. Если is\_word равен 1, то текущий символ-разделитель был после слова, тогда проверяется флаг found, если он равен нулю, то есть условие (две гласные в слове подряд) не выполнено, то текущее слово переписывается в результирующую строку.
6. Если первое условие из пункта 4 не выполняется и предыдущий символ – разделитель, происходит запись начала слова в указатель word\_ptr.
7. Для установления принадлежности буквы слову, написана вспомогательная функция CheckElementInArray.
8. **Схема алгоритма:**

Рисунок 1 - Схема алгоритма

1. **Листинг программы**

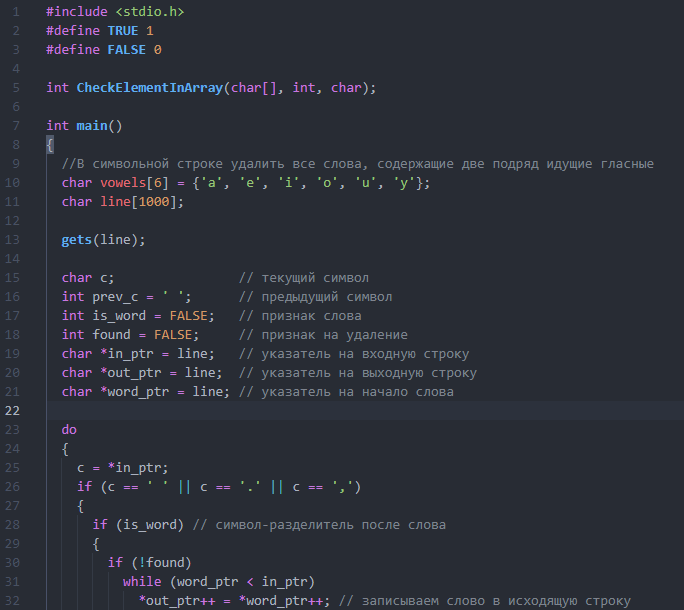


Рисунок 2 – Код программы

Рисунок 2 – Код программы

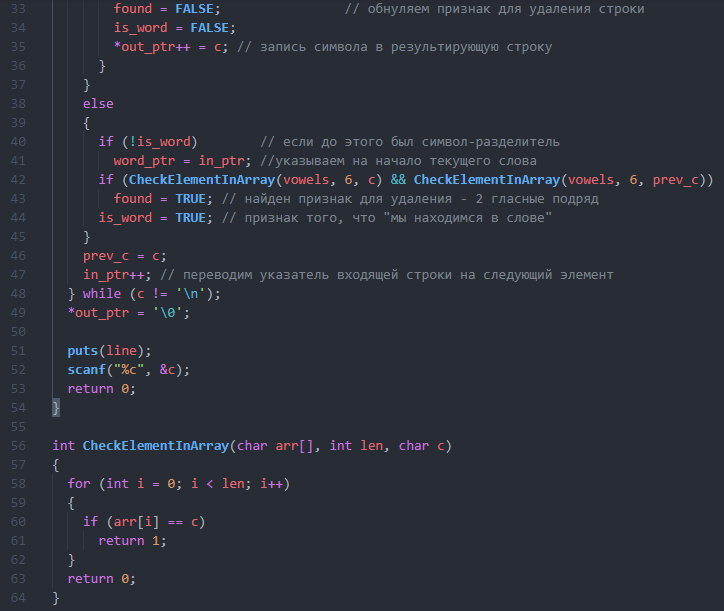


Рисунок 3 – Код программы (продолжение)

1. **Тестовые примеры**

Таблица 1 – Тестовые примеры

|  |  |
| --- | --- |
| **№** | **Тест** |
| 1 |  |
| 2 |  |
| 3 |  |

1. **Вывод**

В результате выполнения лабораторной работы была создана программа, удаляющая в символьной строке все слова, содержащие две подряд идущие гласные буквы.

Выполнив лабораторную работу, я изучил команды адресной арифметики.