ГУАП

КАФЕДРА № 14

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Старший преподаватель		К.Н. Рождественская
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

УКАЗАТЕЛИ

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 1043		Н. А. Смирнов
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2021

1. **Цель работы:**
   Изучение команд адресной арифметики.
2. **Постановка задачи:**
   В символьной строке удалить все слова, содержащие две подряд идущие гласные буквы.
3. **Формализация:**
   1) Исходная строка вводится с клавиатуры.
   2) 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.
   3) 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.
   4) 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.
   5) Если первое условие из пункта 4 не выполняется и предыдущий символ – разделитель, происходит запись начала слова в указатель word_ptr.
   6) Для установления принадлежности буквы слову, написана вспомогательная функция CheckElementInArray.

4. Схема алгоритма:

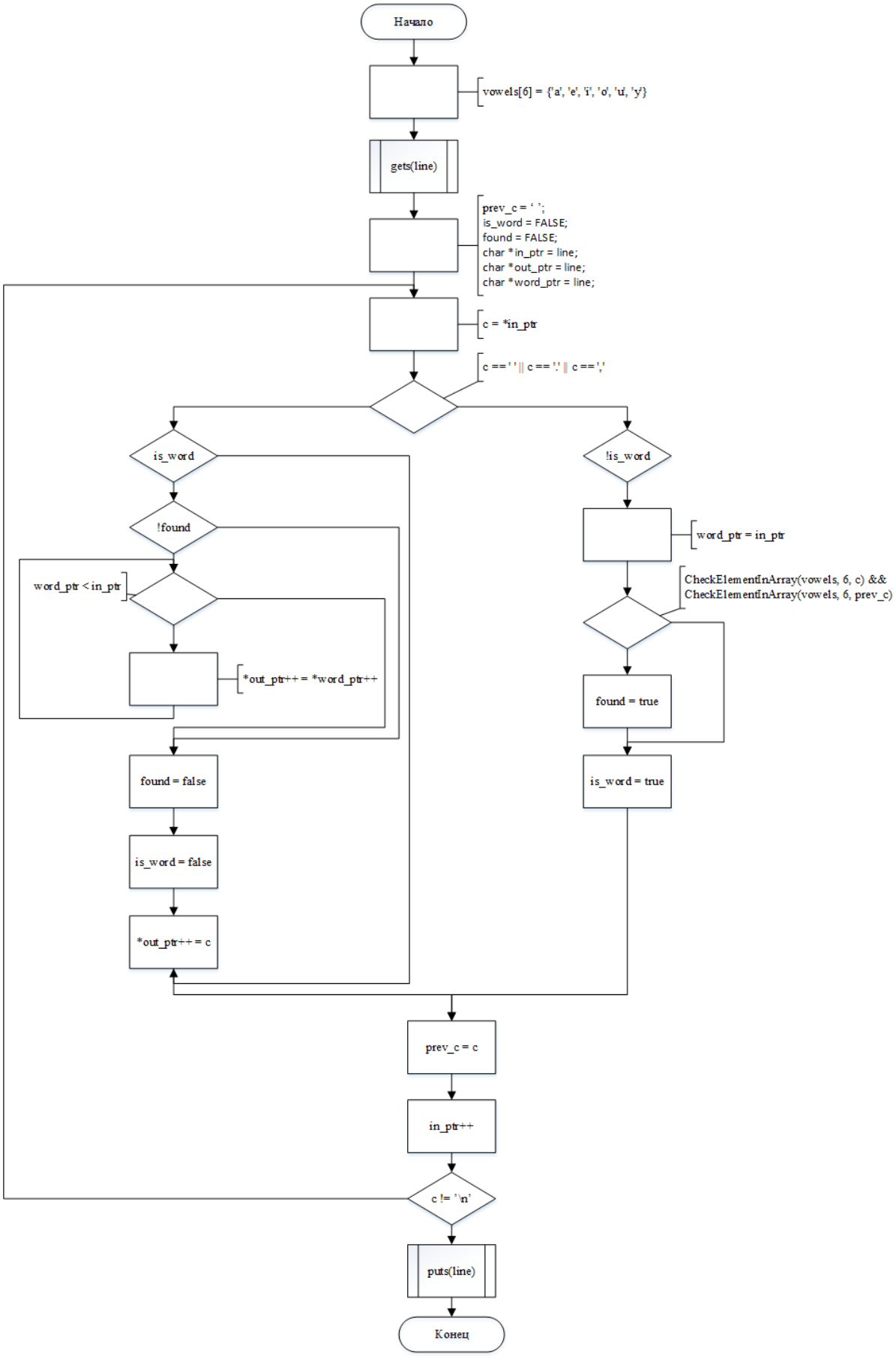


Рисунок 1 - Схема алгоритма

5. Листинг программы

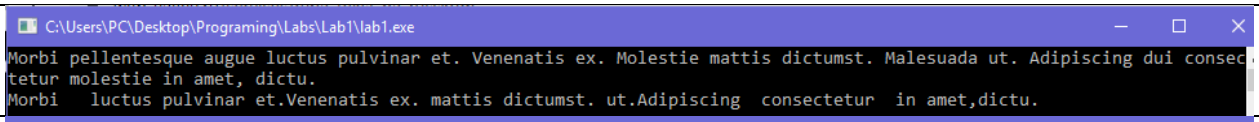
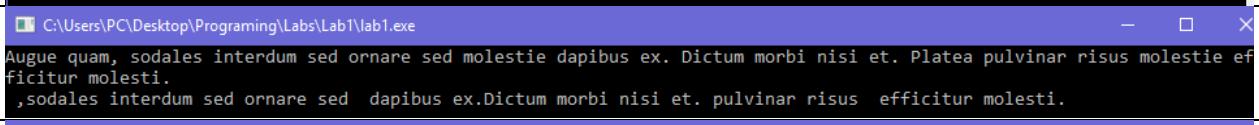
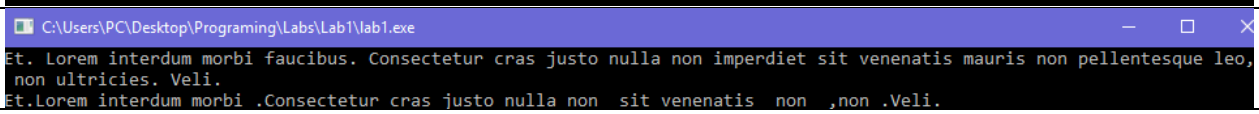
```c
#include <stdio.h>
#define TRUE 1
#define FALSE 0

int CheckElementInArray(char[], int, char);

int main()
{
  //В символьной строке удалить все слова, содержащие две подряд идущие гласные
  char vowels[6] = {'a', 'e', 'i', 'o', 'u', 'y'};
  char line[1000];

  gets(line);

  char c;                // текущий символ
  int prev_c = ' ';      // предыдущий символ
  int is_word = FALSE;   // признак слова
  int found = FALSE;     // признак на удаление
  char *in_ptr = line;   // указатель на входную строку
  char *out_ptr = line;  // указатель на выходную строку
  char *word_ptr = line; // указатель на начало слова

  do
  {
    c = *in_ptr;
    if (c == ' ' || c == '.' || c == ',')
    {
      if (is_word) // символ-разделитель после слова
      {
        if (!found)
          while (word_ptr < in_ptr)
            *out_ptr++ = *word_ptr++; // записываем слово в исходящую строку
        found = FALSE;                // обнуляем признак для удаления строки
        is_word = FALSE;
        *out_ptr++ = c; // запись символа в результирующую строку
      }
    }
    else
    {
      if (!is_word)        // если до этого был символ-разделитель
        word_ptr = in_ptr; //указываем на начало текущего слова
      if (CheckElementInArray(vowels, 6, c) && CheckElementInArray(vowels, 6, prev_c))
        found = TRUE; // найден признак для удаления - 2 гласные подряд
      is_word = TRUE; // признак того, что "мы находимся в слове"
    }
    prev_c = c;
    in_ptr++; // переводим указатель входящей строки на следующий элемент
  } while (c != '\n');
  *out_ptr = '\0';

  puts(line);
  scanf("%c", &c);
  return 0;
}

int CheckElementInArray(char arr[], int len, char c)
{
  for (int i = 0; i < len; i++)
  {
    if (arr[i] == c)
      return 1;
  }
  return 0;
}
```

Рисунок 2 – Код программы

Рисунок 3 – Код программы
(продолжение)

6. Тестовые примеры

Таблица 1 – Тестовые примеры

№	Тест
1	
2	
3	

7. 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.