



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)**

ФАКУЛЬТЕТ
КАФЕДРА

ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

О т ч е т

по домашнему заданию №2(часть 3)
варианта №7

Название лабораторной работы:

Применение множеств

Дисциплина:

Основы программирования

Студент гр. **ИУ6-12**

01.12.16_____
(Подпись, дата)

Векшин Роман
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Черноусова Татьяна Геннадьевна
(И.О. Фамилия)

Задание

Составить программу, используя множественный тип.

Дана последовательность слов из букв латинского алфавита, разделенных пробелами, в конце точка.

Определить, какие согласные буквы не входят ни в одно четное слово этой последовательности. Вывести на экран найденные символы в алфавитном порядке. При отсутствии таких слов выдать соответствующее сообщение.

Схема алгоритма

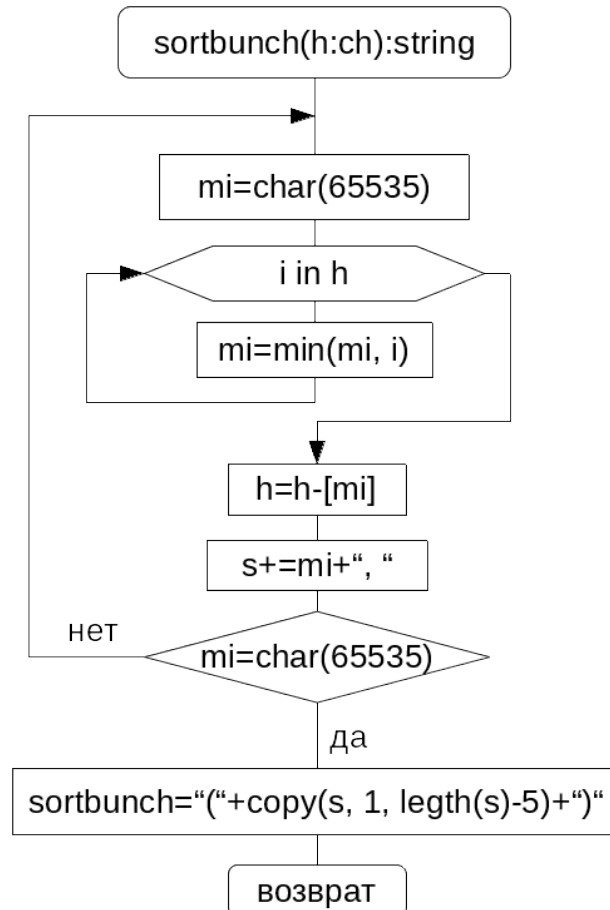


Рисунок 1-Схема алгоритма

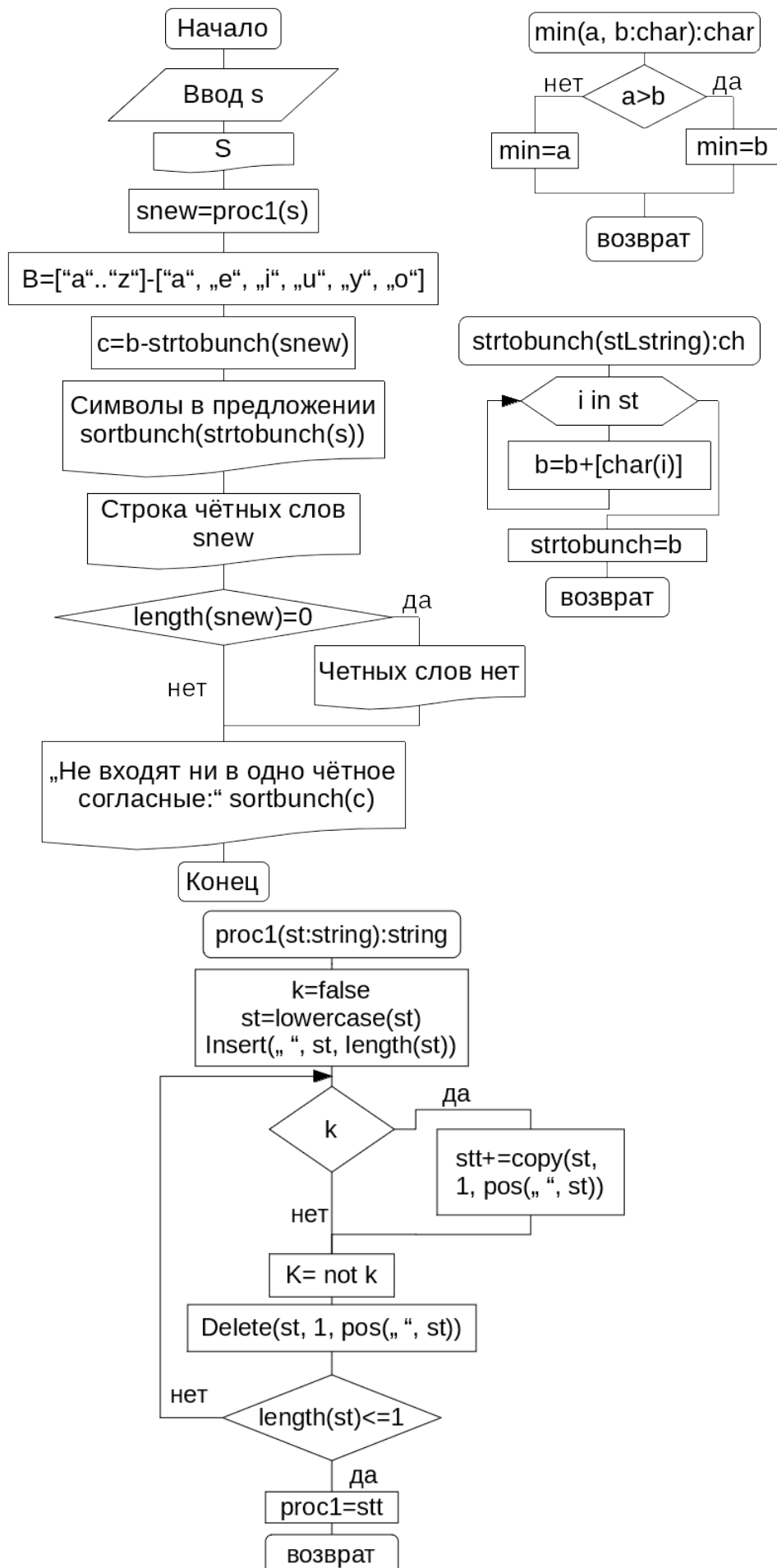


Рисунок 2-Схема алгоритма

Код программы

```
program project1;

{$APPTYPE CONSOLE}

type
  ch = set of char;

  ///Переводим множество в сортированную строку
  function sortbunch(h: ch): string;
  var
    mi, i: char;
    s: string = '';
  begin
    repeat
      mi := char(65535); //Символ с наибольшим кодом
      for i in h do
        if i < mi then
          mi := i;
      h := h - [mi];
      s += mi + ', ';
    until mi = char(65535);
    sortbunch := '(' + copy(s, 1, length(s) - 5) + ')';
  end;

  ///Переводим строку в множество символов
  function StrToBunch(st: string): ch;
  var
    b: set of char = [];
    i: char;
  begin
    for i in st do
      b += [char(i)];
    StrToBunch := b;
  end;

  ///Формируем новую строку из чётных слов
  function proc1(st: string): string;
  var
    stt: string = '';
    k: boolean = False;
  begin
    st := LowerCase(st);
    insert(' ', st, length(st));
    repeat
      if k then
        stt += copy(st, 1, pos(' ', st));
        k := not k;
      Delete(st, 1, pos(' ', st));
    until length(st) <= 1;
    proc1 := stt;
  end;

var
  snew, s: string;
  c, b: ch;

begin
  writeln(utf8toansi('Введите строку '));
  readln(s);
  writeln(utf8toansi('Строка введена. '));
  writeln(utf8toansi('Исходная строка: '), s, ' ');

  snew := proc1(s);
  b := ['a'..'z'] - ['a', 'e', 'i', 'u', 'y', 'o'];
  c := b - StrToBunch(snew);

  writeln(utf8toansi('Найдены символы в предложении: '), sortbunch(StrToBunch(s)));
  writeln(utf8toansi('Строка чётных слов: '), snew, ' ');
  if length(snew) = 0 then
    writeln(utf8toansi('Чётных слов в строке нет!'));
```

```
writeln(utf8toansi('Не входят ни в одно чётное слово согласные:'), sortbunch(c));
writeln(utf8toansi('Нажмите любую клавишу для завершения работы программы.'));
readln();
end
```

Тестирование

Таблица 1-Результаты тестирования программы 1

Исходные данные	Ожидаемый результат	Полученный результат
One can become a writer only if he is talented.	b, f, g, j, k, m, p, q, r, s, v, w, x, z	(b, f, g, j, k, m, p, q, r, s, v, w, x, z)
You have come very fast.	b, c, d, f, g, j, k, l, m, n, p, q, s, t, w, x, z	(b, c, d, f, g, j, k, l, m, n, p, q, s, t, w, x, z)
.	b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, z	(b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, z)
q qwertyuiiopasdfghjkl.	b, c, m, n, v, x, z	(b, c, m, n, v, x, z)
q qwertyuiiopasdfghjklzxcvbnm.	(нет согласных)	()

Пример работы программы

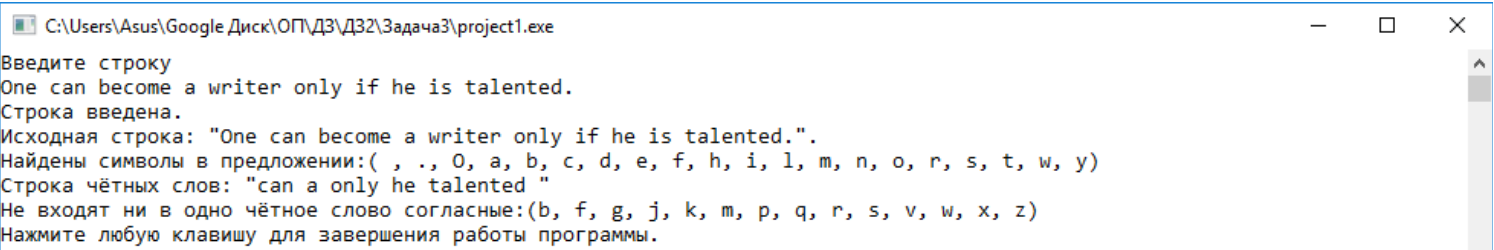


Рисунок 3-Пример работы программы

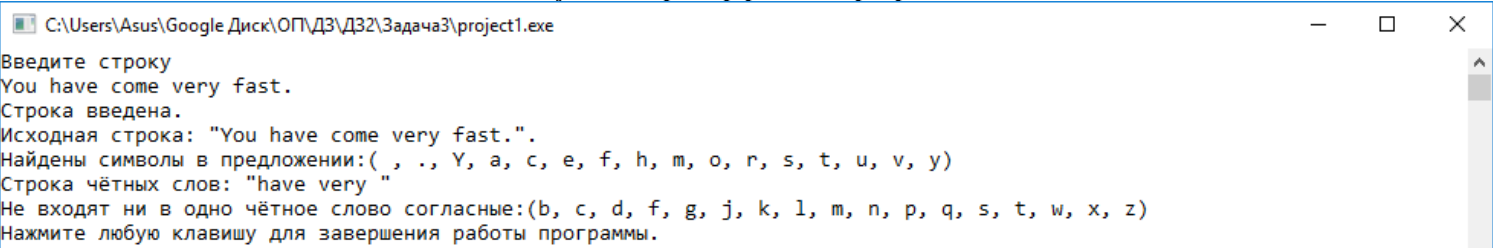


Рисунок 4-Пример работы программы

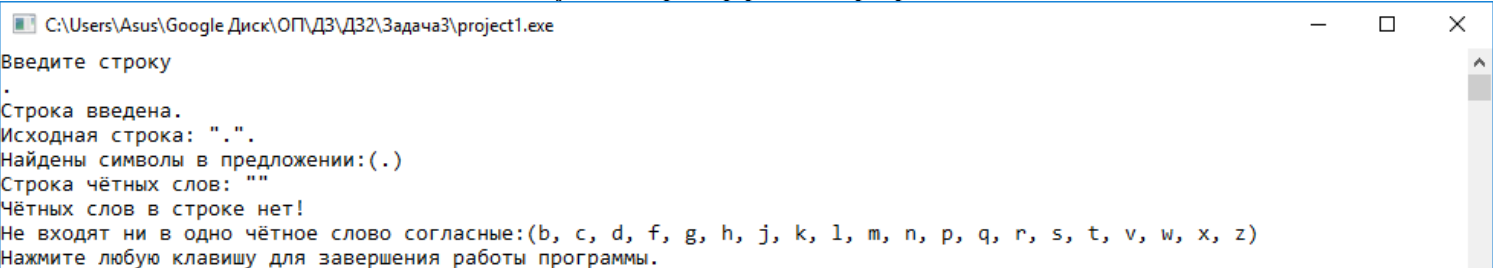


Рисунок 5-Пример работы программы

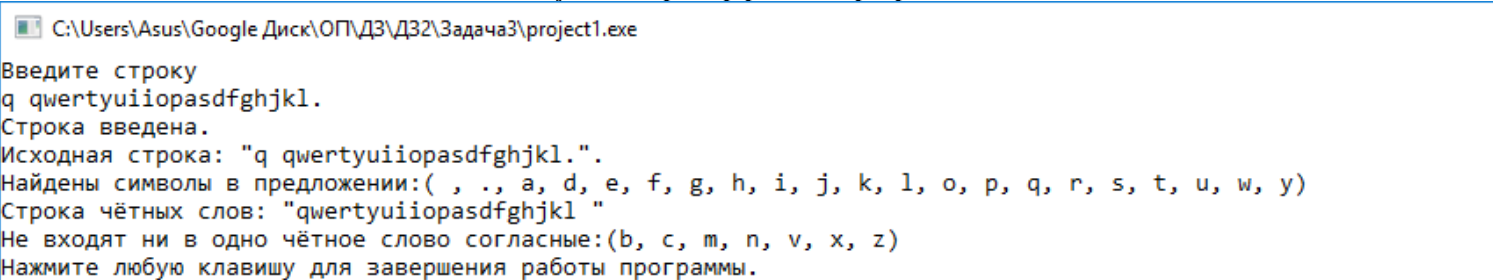


Рисунок 6-Пример работы программы

Рисунок 7-Пример работы программы

Вывод

- 1) Разработан алгоритм поиска чётных слов и согласных, не входящих в эти слова, и составлена его схема в среде LibreOffice Draw (см. рис. 1-2).
- 2) Создан код программы по схеме алгоритма в среде Lazarus.
- 3) Проведено тестирование (см. табл. 1).
- 4) Тестирование показало корректность работы программы (см. рис. 3-7).