|  |  |
| --- | --- |
| Министерство образования Республики Беларусь | |
| Учреждение образования | |
| БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ | |
| ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ | |
|  | |
|  | |
| Факультет компьютерных систем и сетей | |
| Кафедра программного обеспечения информационных технологий | |
| Дисциплина: **Название дисциплины (АББРЕВИАТУРА)** | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
| **ОТЧЁТ** | |
| по лабораторной работе № **X** | |
|  | |
| Тема работы: **Название темы** | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
| Выполнил: | Фамилия И.О. |
|  | гр. **XXXXXX** |
|  | Вариант **X** |
|  |  |
| Проверил: | Фамилия И.О. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Минск **202X** | |

содержание

[1 Постановка задачи 3](#_Toc91177069)

[2 Методика решения 4](#_Toc91177070)

[3 Текстовый алгоритм решения задачи 5](#_Toc91177071)

[4 Структура данных 6](#_Toc91177072)

[5 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc91177073)

[6 Результаты расчетов 9](#_Toc91177074)

[Приложение А 10](#_Toc91177075)

[Приложение Б 12](#_Toc91177076)

[Приложение В 14](#_Toc91177077)

# Постановка задачи

Дана последовательность, содержащая от 2 до 50 слов, в каждом из которых от 1 до 8 строчных латинских букв; между соседними словами – запятая или не менее одного пробела, за последним словом – точка.

Вывести те слова последовательности, которые отличны от последнего слова и удовлетворяют следующему свойству: слово начинается с гласной буквы и заканчивается согласной буквой.

# Методика решения

Для того, чтобы вывести последовательность слов согласно определённому условию, программа должна разделить строку на отдельные слова и удалить оттуда лишние символы (точку, не одинарные пробелы и запятые).

Запятые можно удалить при помощи цикла с предусловием:

*Pos(', ', S) > 0*

Не одинарные пробелы можно удалить при помощи цикла с предусловием:

*Pos(' ', S) > 0*

Точка по условию одна, поэтому её можно удалить, не применяя цикл.

Разделение строки на слова происходит при помощи цикла с предусловием. Каждое слово сохраняется в изначально пустой массив, после чего удаляется из строки, и так до тех пор, пока выполняется условие:

*Pos(' ', S) > 0*

Вывод элементов массива реализуется при помощи третьего цикла с двумя условиями внутри. Первое условие отсеивает слова, являющиеся дубликатами последнего слова. Второе из оставшихся слов выводит те, что удовлетворяют условию «первая буква гласная, последняя согласная».

# Текстовый алгоритм решения задачи

Таблица – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | Ввод S |
|  | S:=S+' ' |
|  | Начало цикла А1. Проверка выполнения условия (Pos(',', S) > 0). Если условие истинно, перейти к шагу 4, иначе – к шагу 6 |
|  | Delete(S, Pos(',', S), 1) |
|  | Конец цикла А1. Идти к шагу 3 |
|  | Начало цикла А2. Проверка выполнения условия (Pos(' ', S) > 0). Если условие истинно, перейти к шагу 7, иначе – к шагу 9 |
|  | Delete(S, Pos(' ', S), 1) |
|  | Конец цикла А2. Идти к шагу 6 |
|  | Delete(S, Pos('.', S), 1) |
|  | N:=1 |
|  | Начало цикла А3. Проверка выполнения условия (Pos(' ', S) > 0) and (N <= 50) . Если условие истинно, перейти к шагу 12, иначе – к шагу 17 |
|  | W:=Copy(S, 1, Pos(' ', S)-1) |
|  | A[N]:= W |
|  | N:=N+1 |
|  | Delete(S, 1, Length(W)+1) |
|  | Конец цикла А3. Идти к шагу 11 |
|  | I:=1 |
|  | Начало цикла А4. Проверка выполнения условия (I <= N-1). Если условие истинно, перейти к шагу 19, иначе – к шагу 26 |
|  | Проверка выполнения условия (A[I]<>A[N-1]). Если условие истинно, перейти к шагу 20, иначе – к шагу 24 |
|  | W:=A[I] |
|  | L:=Length(W) |
|  | Проверка выполнения условия ((W[1] = 'e') or (W[1] = 'y') or (W[1] = 'u') or (W[1] = 'i') or (W[1] = 'o') or (W[1] = 'a') ) and ((W[L] = 'q') or (W[L] = 'w') or (W[L] = 'r') or (W[L] = 't') or (W[L] = 'p') or (W[L] = 's') or (W[L] = 'd') or (W[L] = 'f') or (W[L] = 'g') or (W[L] = 'h') or (W[L] = 'j') or (W[L] = 'k') or (W[L] = 'l') or (W[L] = 'z') or (W[L] = 'x') or (W[L] = 'c') or (W[L] = 'v') or (W[L] = 'b') or (W[L] = 'n') or (W[L] = 'm')). Если условие истинно, перейти к шагу 23, иначе – к шагу 24 |
|  | Вывод A[I] |
|  | I:=I+1 |
|  | Конец цикла А4. Идти к шагу 18 |
|  | Останов. |

# Структура данных

Таблица – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| S | String | Строка |
| W | String | 1 слово |
| A | Array[1..50] of String | Массив со словами |
| N | Integer | Количество слов |
| I | Integer | Параметр цикла |
| L | Integer | Длина слова |
| Flag | Boolean | Существуют ли слова |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

|  |
| --- |
|  |
| Рисунок 1 – Схема алгоритма решения задачи по ГОСТ 19.701-90 |

|  |
| --- |
|  |
| Рисунок 2 – Схема алгоритма решения задачи по ГОСТ 19.701-90 |

# Результаты расчетов

Вследствие результатов программы на экран выводятся следующие результаты расчетов:

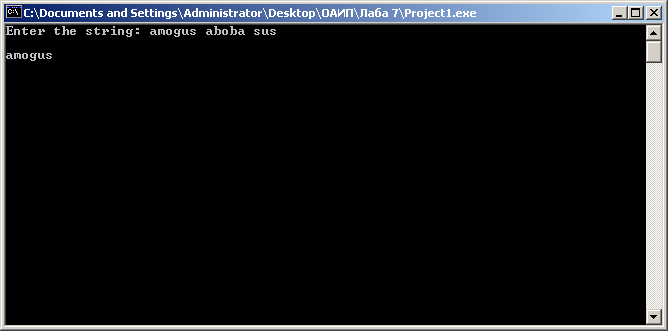


Рисунок 3 – Результаты расчетов

Приложение А

(обязательное)

Исходный код программы (Реализация 1)

Program Lab7;

{This program returns words with conditions: first letter is vowel, last is conent and the word differs from the last entered word}

//Use app

{$APPTYPE CONSOLE}

//Declare modules

Uses

SysUtils;

//Declare vars

Var

S, W: String;

A: Array[1..50] of String;

N, I, L: Integer;

Flag: Boolean;

//S - string

//W - word

//A - array with words

//N - array length

//I - cycle parameter

//L - word length

//Flag - boolean about existance

Begin

Write('Enter the string: ');

ReadLn(S);

//To not remove last word

S:= S + ' ';

//Remove unneeded symbols

While Pos(',', S) > 0 do

Delete(S, Pos(',', S), 1);

While Pos(' ', S) > 0 do

Delete(S, Pos(' ', S), 1);

Delete(S, Pos('.', S), 1);

WriteLn;

N:= 1;

//Find words

While (Pos(' ', S) > 0) and (N <= 50) do

Begin

W:= Copy(S, 1, Pos(' ', S) - 1); (\* Find the word \*)

A[N]:= W; (\* Save the word \*)

N:= N + 1; (\* Increment N \*)

Delete(S, 1, Length(W) + 1); (\* Remove word, space \*)

End;

//Zero words yet

Flag:= True;

//Find needed words

For I:= 1 to N - 1 do

Begin

//Condition about last item

If A[I] <> A[N - 1] then

Begin

W:= A[I]; (\* Load saved word \*)

L:= Length(W);

//Condition about letters

If ((W[1] = 'e') or (W[1] = 'y') or (W[1] = 'u') or (W[1] = 'i') or (W[1] = 'o') or (W[1] = 'a') ) and ((W[L] = 'q') or (W[L] = 'w') or (W[L] = 'r') or (W[L] = 't') or (W[L] = 'p') or (W[L] = 's') or (W[L] = 'd') or (W[L] = 'f') or (W[L] = 'g') or (W[L] = 'h') or (W[L] = 'j') or (W[L] = 'k') or (W[L] = 'l') or (W[L] = 'z') or (W[L] = 'x') or (W[L] = 'c') or (W[L] = 'v') or (W[L] = 'b') or (W[L] = 'n') or (W[L] = 'm')) then

Begin

Flag:= False; (\* Has results \*)

Write(A[I], ' '); (\* Displaying the word \*)

End;

End;

End;

//If has no results

If Flag then

Writeln('There is no needed words.');

Readln;

End.

Приложение Б

(обязательное)

Исходный код программы (Реализация 2)

Program Lab7;

{This program returns words with conditions: first letter is vowel, last is conent and the word differs from the last entered word}

//Use app

{$APPTYPE CONSOLE}

//Declare modules

Uses

SysUtils;

//Declare consts

Const

Vow = ['e', 'y', 'u', 'i', 'o', 'a'];

Con = ['q', 'w', 'r', 't', 'p', 's', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'z', 'x', 'c', 'v', 'b', 'n', 'm'];

//Declare vars

Var

S, W: String;

A: Array[1..50] of String;

N, I, L: Integer;

Flag: Boolean;

//S - string

//W - word

//A - array with words

//N - array length

//I - cycle parameter

//L - word length

//Flag - boolean about existance

Begin

Write('Enter the string: ');

ReadLn(S);

//To not remove last word

S:= S + ' ';

//Remove unneeded symbols

While Pos(',', S) > 0 do

Delete(S, Pos(',', S), 1);

While Pos(' ', S) > 0 do

Delete(S, Pos(' ', S), 1);

Delete(S, Pos('.', S), 1);

WriteLn;

N:= 1;

//Find words

While (Pos(' ', S) > 0) and (N <= 50) do

Begin

W:= Copy(S, 1, Pos(' ', S) - 1); (\* Find the word \*)

A[N]:= W; (\* Save the word \*)

N:= N + 1; (\* Increment N \*)

Delete(S, 1, Length(W) + 1); (\* Remove word, space \*)

End;

//Zero words yet

Flag:= True;

//Find needed words

For I:= 1 to N - 1 do

Begin

//Condition about last item

If A[I] <> A[N - 1] then

Begin

W:= A[I]; (\* Load saved word \*)

L:= Length(W);

//Condition about letters

If (W[1] in Vow) and (W[L] in Con) then

Begin

Flag:= False; (\* Has results \*)

Write(A[I], ' '); (\* Displaying the word \*)

End;

End;

End;

//If has no results

If Flag then

Writeln('There is no needed words.');

Readln;

End.

Приложение В

(обязательное)

Тестовые наборы

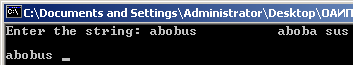
Тестовая ситуация 1: проверка вывода

Тестовая ситуация: проверка, выведет ли программа слова, если строка содержит более двух пробелов, запятые или точку.

Исходные данные: S = abobus aboba sus

Ожидаемый результат: abobus

Полученный результат:



Исходные данные: S = amomus, amoma, sus.

Ожидаемый результат: amomus

Полученный результат:



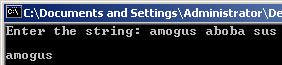
Тестовая ситуация 2: проверка результата

Тестовая ситуация: проверка, выведет ли программа слова, либо сообщение, что их нет.

Исходные данные: S = amogus aboba sus

Ожидаемый результат: amogus

Полученный результат:



Исходные данные: S = оаоаоа ммм

Ожидаемый результат: There is no needed words.

Полученный результат:



Исходные данные: S = do you like deep dark fantasies for three hundred bucks

Ожидаемый результат: There is no needed words.

Полученный результат:

