Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по лабораторной работе № 7

Тема работы: Строки(отчет 2)

Выполнил

студент: гр. 151003 Барановский Р.А.

Проверил: Фадеева Е.П.

Минск 2021

содержание

[1 Постановка задачи 3](#_Toc90311985)

[2 Текстовый алгоритм решения задачи 4](#_Toc90311986)

[3 Структура данных 6](#_Toc90311987)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc90311988)

[5 Результаты расчетов 11](#_Toc90311989)

[Приложение А 12](#_Toc90311990)

[Приложение Б 16](#_Toc90311991)

# 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.

Напечатать те слова последовательности, которые отличны от последнего слова и удовлетворяют следующему свойству:

* в слове нет повторяющихся букв

# Текстовый алгоритм решения задачи

Таблица – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
| 1 | Ввод Str |
| 2 | Temp := ‘’ |
| 3 | I := length(Str) – 1 |
| 4 | Начало цикла А1. Проверка выполнения условия (Str[I] < > ‘,’) and (Str[I] < > ‘ ‘). Если условие истинно, то идти к шагу 5, иначе к шагу 8 |
| 5 | Temp := Temp + copy(Str, I, 1) |
| 6 | dec(I) |
| 7 | Конец цикла А1. Идти к шагу 4 |
| 8 | LastWord := ‘’ |
| 9 | I := 1 |
| 10 | Начало цикла А2. Проверка выполнения условия I <= length(Temp). Если условие истинно, то идти к шагу 11, иначе к шагу 14 |
| 11 | LastWord := Temp[I] + LastWord |
| 12 | I := I + 1 |
| 13 | Конец цикла А2. Идти к шагу 10 |
| 14 | Вывод Str |
| 15 | Вывод LastWord |
| 16 | I := 1 |
| 17 | K := 1 |
| 18 | Начало цикла А3. Проверка выполнения условия Str[I] < > ‘.’. Если условие истинно, то идти к шагу 19, иначе к шагу 43 |
| 19 | Проверка выполнения условия (Str[I] < > ‘ ‘) and (Str[I] < > ‘,’). Если условие истинно, то идти к шагу 20, иначе к шагу 41 |
| 20 | Checker := true |
| 21 | Word := ‘’ |
| 22 | Начало цикла А4. Проверка выполнения условия (Str[I] < > ‘ ‘) and (Str[I] < > ‘,’) and (I < length(Str)). Если условие истинно, то идти к шагу 23, иначе к шагу 26 |
| 23 | Word := Word + Str[I] |
| 24 | inc(I) |
| 25 | Конец цикла А4. Идти к шагу 22 |
| 26 | MasStr[K] := Word |
| 27 | WordCopy := MasStr[K] |

Продолжение таблицы 1

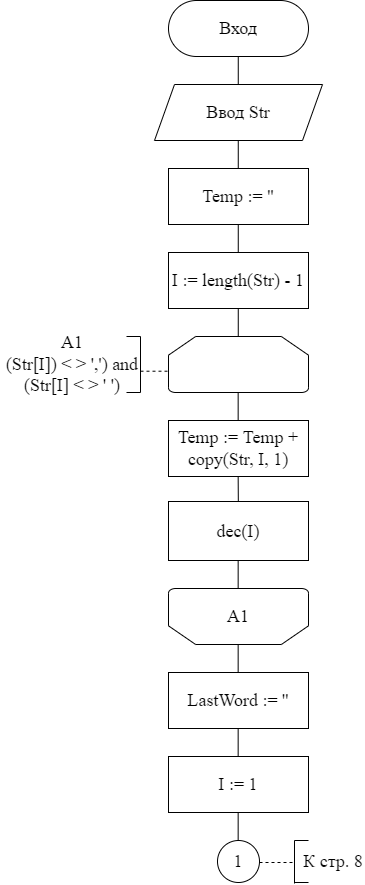
|  |  |
| --- | --- |
| 28 | J := 1 |
| 29 | Начало цикла А5. Проверка выполнения условия J <= length(Word). Если условие истинно, то идти к шагу 30, иначе к шагу 36 |
| 30 | Letter := copy(MasStr[K], J, 1) |
| 31 | delete(WordCopy, 1, 1) |
| 32 | Проверка выполнения условия pos(Letter,WordCopy) < > 0. Если условие истинно, то идти к шагу 33, иначе к шагу 34 |
| 33 | Checker := false |
| 34 | J := J + 1 |
| 35 | Конец цикла А5. Идти к шагу 29 |
| 36 | Проверка выполнения условия (MasStr[K] = LastWord) or (Checker = false). Если условие истинно, то идти к шагу 37, иначе к шагу 39 |
| 37 | MasStr[K] := ‘’ |
| 38 | Конец условия (MasStr[K] = LastWord) or (Checker = false). Идти к шагу 40 |
| 39 | inc(K) |
| 40 | Конец условия (Str[i] < > ‘ ‘) and (Str[I] < > ‘,’). Идти к шагу 42 |
| 41 | inc(I) |
| 42 | Конец цикла А3. Идти к шагу 18 |
| 43 | I := 1 |
| 44 | Начало цикла А6. Проверка выполнения условия I <= K-1. Если условие истинно, то идти к шагу 45, иначе к шагу 48 |
| 45 | Вывод MasStr[I] |
| 46 | I := I + 1 |
| 47 | Конец цикла А6. Идти к шагу 44 |
| 48 | Останов. |

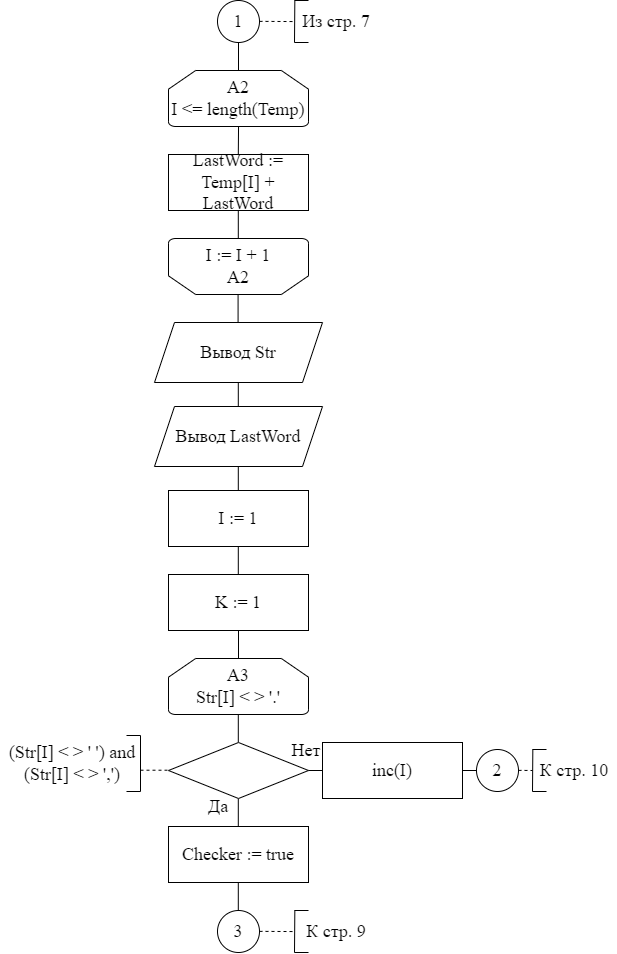
# Структура данных

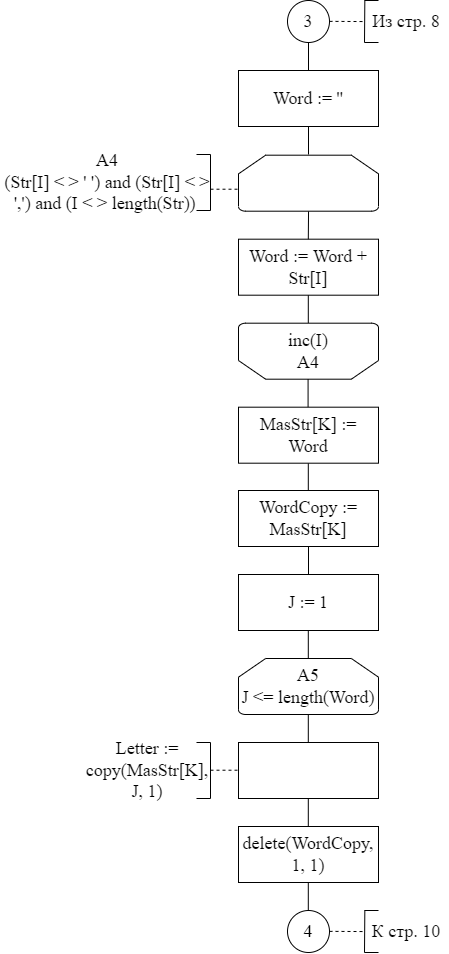
Таблица - Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| Mas | Array of String | Массив с словами |
| MasStr | Mas | Массив с словами |
| Str | String | Изначальная строка |
| LastWord | String | Последнее слово строки |
| Word | String | Текущее проверяемое слово строки |
| Letter | String | Переменная, содержащая 1-ый, 2-ой и т.д. символы Word для того, чтобы узнать, есть ли в слове повторяющиеся буквы |
| WordCopy | String | Копия Word |
| Temp | String | Переменная, содержащая палиндром последнего слова строки |
| I | Integer | Счетчик 1 |
| J | Integer | Счетчик 2 |
| K | Integer | Счетчик 3 |
| Checker | Boolean | Используется для проверки ошибок ввода и затем принимает значение false, если в проверяемом слове есть повторяющиеся буквы, true, если нет |

# Схема алгоритма решения задачи по ГОСТ 19.701-90







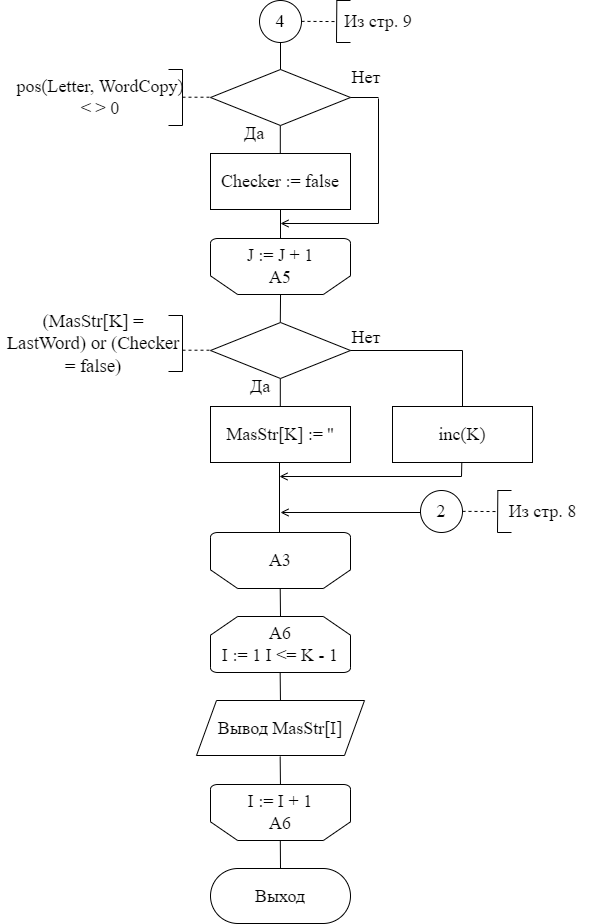


Рисунок - Схема алгоритма решения задачи по ГОСТ 19.701-90

# Результаты расчетов

Вследствие результатов программы на экран выводятся следующие результаты расчетов:

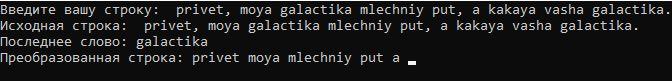


Рисунок – результаты расчетов

Приложение А

(обязательное)

Исходный код программы

Program Exp2;

//Program removes odd elements

//Console app

{$APPTYPE CONSOLE}

{$R \*.res}

//Modules declaration

Uses

System.SysUtils;

Type

Mas = array[1..50] of string;

//Variables declaration

Var

Str, LastWord: string;

Word, Letter, WordCopy, Temp: string;

MasStr: Mas;

I, J, K: integer;

Checker: boolean;

{ Str - original string

LastWord - the last word

Word - current checking word

Letter - 1-st, 2-nd, etc. letter of Word

WordCopy - copy of Word

Temp - variable, which contains palindrom of LastWord

MasStr - array with words

I - counter 1

J - counter 2

K - counter 3

Checker - variable, which is used to find mistakes and to

delete words with repeating letters }

//Start the program

Begin

//Mistakes checker

repeat

//Set checker to the original value

Checker := true;

//Input String

write('Введите вашу строку: ');

readln(Str);

//Check condition of only english letters,

//allowing input spaces and сommas

if length(Str) <> 0 then

begin

for I := 1 to length(Str) - 1 do

begin

if ((ord(Str[I]) < 96) or (ord(Str[I]) > 125)) and

(Str[I] <> ' ') and (Str[I] <> ',') then

Checker := false;

end;

end

//If nothing was inputted

else

Str := ' ';(\*In order to make insufficient input\*);

//If mistake was found

if (Str = '') or ((pos(' ',Str) = 0) and

(pos(',',Str) = 0)) or (Checker = false) or

(Str[length(Str)] <> '.') then

writeln('Ошибка ввода');

//Check if input is sufficient

until (Checker = true) and (Str[length(Str)] = '.') and

(Str <> '') and ((pos(' ',Str) <> 0) or

(pos(',',Str) <> 0));

//Set Temp to the original value

Temp := '';

//Set i to the original value

I := length(Str) - 1;

//Find palindrom of the last word

while (Str[I] <> ',') and (Str[I] <> ' ') do

begin

Temp := Temp + copy(Str,I,1);

dec(I);

end;

//Set LastWord to the original value

LastWord := '';

//Write palindrom of the last word in reverse order

for I := 1 to length(Temp) do

LastWord := Temp[I] + LastWord;

//Output original string and last word

writeln('Исходная строка: ', Str);

writeln('Последнее слово: ', LastWord);

//Set I to the original value

I := 1;

//Set K to the original value

K := 1;

//While Str[i] not equals '.'

while Str[I] <> '.' do

begin

//If new word was found

if (Str[I] <> ' ') and (Str[I] <> ',') then

begin

//Set checker to the original value

Checker := true;

//Set Word to the original value

Word := '';

//Find Word

while (Str[I] <> ' ') and (Str[I] <> ',') and

(I < Length(Str)) do

begin

Word := Word + Str[I];

inc(I);

end;

//Fill the array

MasStr[K] := Word;

//Set WordCopy to the original value

WordCopy := MasStr[K];

//Check if there are repeated letters in Word

for J := 1 to length(Word) do

begin

//Copy 1-st, 2-nd, ... letter of the word

Letter := Copy(MasStr[K],J,1);

//Delete this letter

delete(WordCopy,1,1);

//Check if the same letter was found

if pos(Letter,WordCopy) <> 0 then

Checker := false;

end;

//If found word equals to the last word or there

//are repeated lettters

if (MasStr[K] = LastWord) or (Checker = false) then

MasStr[K] := '';

//If word corresponds to all conditions

else

inc(K);

end

//If we met space or comma

else

inc(I);

end;

//Output transformed array

write('Преобразованная строка: ');

for I := 1 to K-1 do

write(MasStr[I],' ');

//Final operations

readln;

End.

Приложение Б

(обязательное)

Тестовые наборы(Расчеты)

Тест 1

Тестовая ситуация: Стандартный режим работы программы

Исходные данные: Str = ‘pogib poet, nevolnik chesti, pal,

oklevetaniy molvoy.’

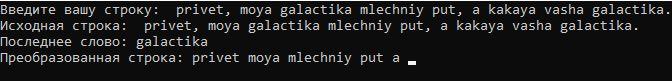
Ожидаемый результат:

Исходная строка: pogib poet, nevolnik chesti, pal, oklevetaniy molvoy.

Последнее слово: molvoy

Преобразованная строка: pogib poet chesti pal

Полученный результат:



Тест 2

Тестовая ситуация: Стандартный режим работы программы

Исходные данные: Str = ‘britanskiy kot samiy lucshiy kot.’

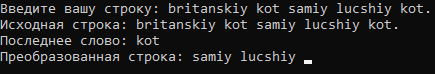
Ожидаемый результат:

Исходная строка: britanskiy kot samiy lucshiy kot.

Последнее слово: kot

Преобразованная строка: samiy lucshiy

Полученный результат:



Тест 3

Тестовая ситуация: Стандартный режим работы программы

Исходные данные: Str = ‘odin, dva, tri, chetire, odin.’

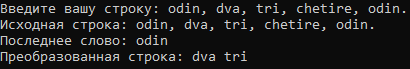
Ожидаемый результат:

Исходная строка: odin, dva, tri, chetire, odin.

Последнее слово: odin

Преобразованная строка: dva tri

Полученный результат:



Тест 4

Тестовая ситуация: Стандартный режим работы программы

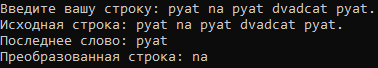
Исходные данные: Str = ‘pyat na pyat dvadcat pyat.’

Ожидаемый результат:

Исходная строка: pyat na pyat dvadcat pyat.

Последнее слово: pyat

Преобразованная строка: na

Полученный результат:

Тест 5

Тестовая ситуация: Все слова одинаковые

Исходные данные: Str = ‘one one one one one one.’

Ожидаемый результат:

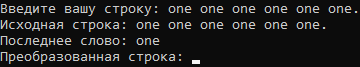
Введите вашу строку: one one one one one one.

Исходная строка: one one one one one one.

Последнее слово: one

Преобразованная строка:

Полученный результат:



Тест 6

Тестовая ситуация: Все слова разные

Исходные данные: Str = ‘yes, no, vip, das, ist.’

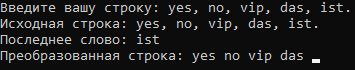
Ожидаемый результат:

Введите вашу строку: yes, no, vip, das, ist.

Исходная строка: yes, no, vip, das, ist.

Последнее слово: ist

Преобразованная строка: yes no vip das

Полученный результат:

Тестовые наборы(Проверка ввода)

Тест 7

Тестовая ситуация: В строке есть какие-либо символы помимо строчных латинских букв, запятых, пробелов, точки на конце.

Исходные данные:

Str = ‘я люблю мороженое.’

Ожидаемый результат:

Ошибка ввода

Полученный результат:



Тест 8

Тестовая ситуация: На конце предложения отсутствует точка

Исходные данные: Str = ‘never gonna give you up, never gonna let you

down’

Ожидаемый результат:

Ошибка ввода

Полученный результат:

