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:

Работа с строковыми типами данных

Выполнил

студент: гр. 151004 Башлыков В.В.

Проверил: Фадеева Е. П.

Минск 2021

Содержание

[1 Постановка задачи 3](#_Toc88834458)

[2 Ход решения 4](#_Toc88834459)

[3 Текстовый алгоритм решения задачи 5](#_Toc88834460)

[4 Структура данных 8](#_Toc88834461)

[5 Схема алгоритма решения задачи по ГОСТ 19.701-90 9](#_Toc88834462)

[6 Результаты расчетов 14](#_Toc88834463)

[Приложение А 15](#_Toc88834464)

[Приложение Б 21](#_Toc88834465)

[Приложение В 25](#_Toc88834466)

[Приложение Г 31](#_Toc88834467)

[Приложение Д 37](#_Toc88834468)

# Постановка задачи

Дается последовательность, содержащая от 2 до 50 слов, в каждом из которых от 1 до 50 строчных латинских букв; между соседними словами - запятая или хотя бы один пробел, после последнего слова - точка. Выведите те слова последовательности, которые отличаются от последнего слова и удовлетворяют следующему свойству:

- слово симметрично

Правила постановки:

1. Программа учитывает некорректный ввод пользователя.

* Пустая строка
* Отсутствие пробелов
* Отсутствие запятых
* Неподходящих символов

1. Для решения задачи используется строковый тип данных.

# Ход решения

1. В начале необходимо получить последнее слово последовательности. Для этого мы идем с последнего элемента последовательности до момента встречи первой запятой или пробела.
2. После проходя по строке, получаем каждое слово по отдельности.
3. Переворачиваем это слово.
4. Провереяем текущее слово с перевернутым текущим словом и узнаем полиндром оно или нет.
5. Если слово полиндром и не равно последнему слову, тогда выводим слово.

# Текстовый алгоритм решения задачи

Таблица 1 – Алгоритм решения

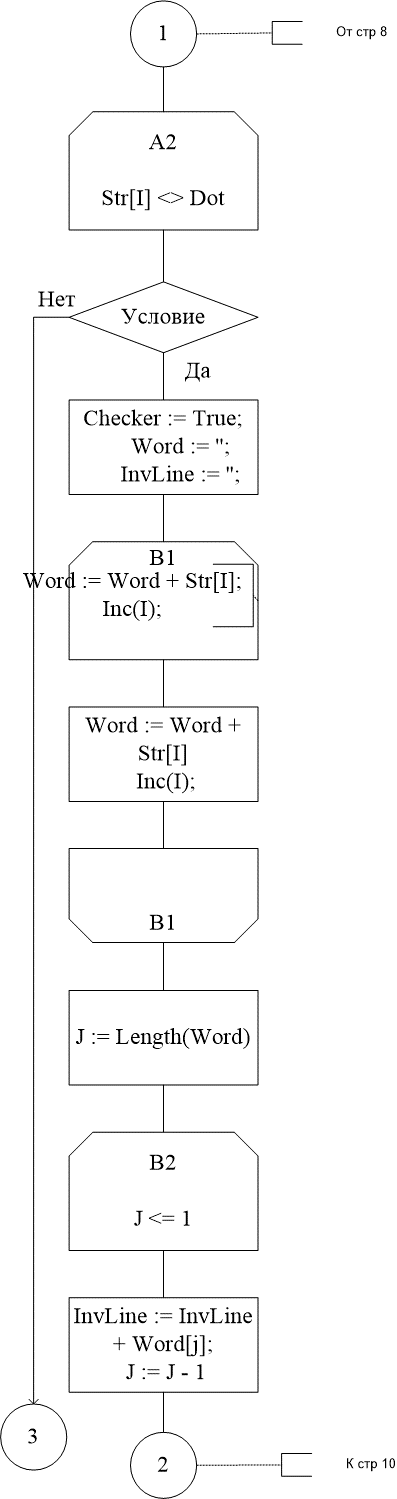
|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | Ввод Str |
|  | I := Length(Str) - 1 |
|  | Начало цикла A1. Проверка выполнения условия (Str[I] <> Comma) and (Str[I] <> Whitespace). Если условие истинно – идти к шагу 4, иначе - идти к шагу 7 |
|  | Temp := Temp + copy(Str, I, 1) |
|  | I := I – 1 |
|  | Конец цикла A1. Идти к шагу 3 |
|  | LastWord := '' |
|  | Начало цикла A1.1. Проверка выполнения условия I <= Length(Temp). Если условие истинно – идти к шагу 9, иначе идти к шагу 11 |
|  | LastWord := Temp[I] + LastWord |
|  | Конец цикла A1.1. Идти к шагу 8 |
|  | I := 1 |
|  | Начало цикла A2. Проверка выполнения условия Str[I] <> Dot.  Если условие истинно – идти к шагу 13, инече – идти к шагу |
|  | Проверка выполнения условия (Str[I] <> Whitespace) and (Str[I] <> Comma). Если условие истинно – идти к шагу 14, иначе - идти к шагу 31 |
|  | Checker := True |
|  | Word := ‘’ |
|  | InvLine := ‘’ |
|  | Начало цикла В1. Проверка выполнения условия (Str[I] <> Whitespace) and (Str[I] <> Comma) and (I < Length(Str)). Если условие истинно – идти к шагу 18, иначе – идти к шагу 21 |
|  | Word := Word + Str[I] |
|  | I := I + 1 |
|  | Конец цикла B1. Идти к шагу 17 |
|  | J := Length(Word) |
|  | Начало цикла В2. Проверка выплнения условия J <= 1. Если условие истинно – идти к шагу 23, иначе – идти к шагу 25 |
|  | InvLine := InvLine + Word[j] |
|  | Конец цикла В2. Идти к шагу 22 |
|  | J := Length(Word) |
|  | Начало цикла В3. Проверка выполнения условия J <= 1. Если условие истинно – идти к шагу 27, иначе мдти к шагу 30 |
|  | Проверка выполнения условия Word[j] <> InvLine[j]. Если условие истинно идти к шагу 28, иначе идти к шагу 29 |
|  | Checker := False; |
|  | Проверка выполнения условия (Word <> LastWord) and (Checker = true). Если условие истинно – идти к шагу 30, иначе – идти к шагу 31 |
|  | Вывод Word |
|  | I := I + 1 |
|  | Конец цикла А2. Идти к шагу 12 |
|  | Останов. |

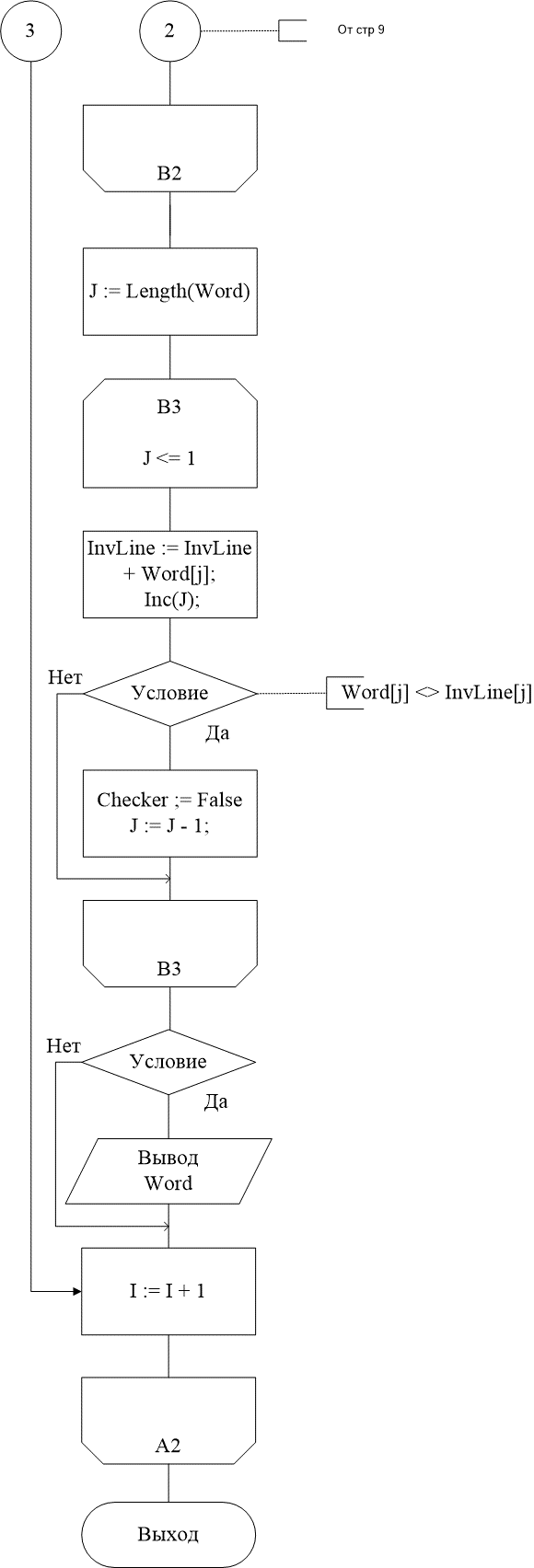
# Структура данных

Таблица 3 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| Str | String | Вводимая строка |
| LastWord | String | Последнее слово строки |
| Word | String | Отдельно взятое слово |
| Temp |  | Переменная для хранения промежуточного результата |
| InvLine | String | Перевернутая строка |
| I | Integer | Счетчик цикла |
| J | Integer | Счетчик цикла |

# Схема алгоритма решения задачи по ГОСТ 19.701-90





# Результаты расчетов

Вследствие выполнения программы на экран выводятся следующие результаты:

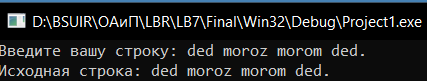


Рисунок 6 – Введё нные данные

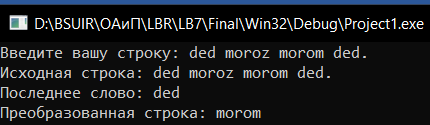


Рисунок 7 – Полученный результат

Приложение А

(обязательное)

Исходный код программы (постановка 1)

program Project1;

{A sequence is given containing from 2 to 50 words, in each of which from 1 to b lowercase Latin letters; between adjacent words - a comma or at least one space, after the last word - a dot. Print those words of the sequence that are different from the last word and satisfy the following property: - the word is symmetrical}

{$APPTYPE CONSOLE}

{$R \*.res}

Uses

System.SysUtils;

Const

Dot = '.';

Whitespace = ' ';

Comma = ',';

//Dot - dor/point

//Whitespace - whitespace

//Comma - Comma

Var

Str, LastWord: String;

Word, Temp, InvLine: String;

I, J: integer;

Checker: boolean;

//Str - first string

//LastWord - Last word in the first string

//Word - single word

//Temp - value from getting the last last word

//InvLine - value from inverted word

//I - counter of the cycle

//J - counter of the cycle

//Checker - Check from getting an answer

Begin

//Start of the cycle C1

Repeat

Checker := True;

Write('Введите вашу строку: ');

ReadLn(Str);

//Check Length(Str) <> 0

if Length(Str) <> 0 then

begin

//Start of the cycle D1

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

begin

//Check correct/uncorrect input

if ((ord(Str[I]) < 96) or (ord(Str[I]) > 122))

and (Str[I] <> Whitespace) and (Str[I] <> Comma)

then

Checker := False;

//End of the cycle D1

end;

end

else Str := Whitespace;

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

(pos(Comma, Str) = 0)) or (Checker = False)

or (Str[length(Str)] <> Dot) then

writeln('Ошибка ввода');

//End of the cycle C1

Until (Checker = True) and (Str[length(Str)] = Dot)

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

or (pos(Comma, Str) <> 0));

Temp := '';

I := Length(Str) - 1;

//Start of the cycle A1

//Search the last word

while (Str[I] <> Comma) and (Str[I] <> Whitespace) do

begin

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

Dec(I);

end;

LastWord := '';

//Start of the cycle A1.1

for I := 1 to length(Temp) do

LastWord := Temp[I] + LastWord;

WriteLn('Исходная строка: ', Str);

WriteLn('Последнее слово: ', LastWord);

Write('Преобразованная строка: ');

I := 1;

//Start of the sycle A2

while Str[I] <> Dot do

begin

//Checking (Str[I] <> Whitespace) and (Str[I] <> Comma)

if (Str[I] <> Whitespace) and (Str[I] <> Comma) then

begin

Checker := True;

Word := '';

InvLine := '';

//Start of the cycle B1

//getting a single word

while (Str[I] <> Whitespace) and (Str[I] <> Comma)

and (I < Length(Str)) do

begin

Word := Word + Str[I];

Inc(I);

//End of the cycle B1

end;

//Start of the cycle B2

//Getting a inverted word

for J := Length(Word) downto 1 do

begin

InvLine := InvLine + Word[j];

//End of the cycle B2

end;

//Start of the cycle B3

//Chaeck from polindrom

for J := Length(Word) downto 1 do

Begin

if Word[j] <> InvLine[j] then Checker := False;

//End of the cycle B3

End;

//Check from polindrom

if (Word <> LastWord) and (Checker = true) then

write(Word, Whitespace);

end

else inc(I);

//End of the cycle A2

end;

ReadLn;

end.

end.end.Приложение Б

(обязательное)

Тестовые наборы

Тестовая ситуация 1 для проверки ввода данных

Тест 1

Тестовая ситуация для проверки, сообщит ли программа о некорректном вводе, если введен элемент массива больше допустимого значения Integer.

Исходные данные: X[1]: 99999999999

Ожидаемый результат: Incorrect value.

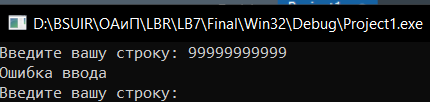
Полученный результат:

Рисунок 8 **–** Полученный результат, тест 1

Тест 2

Тестовая ситуация для проверки, сообщит ли программа о некорректном вводе, если введен не целочисленный элемент массива.

Исходные данные: X[1]: 1.1

Ожидаемый результат: Incorrect value.

Полученный результат:



Рисунок 10 – Полученный результат, тест 3

Тестовая ситуация 2 для проверки вычислений

Тест 1

Тестовые ситуации для проверки вывода правильного решения.

Исходные данные:

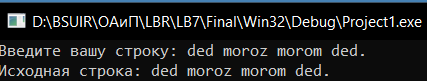


Рисунок 11 **–** Исходные данные, тест 1

Ожидаемый результат: morom

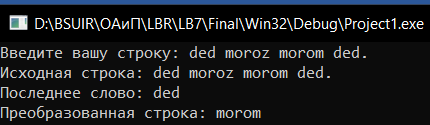
Полученный результат:

Рисунок 12 – Полученный результат, тест 1