Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

# оТЧЕТ

по лабораторной работе

на тему:

Изучение строкового типа данных

|  |  |  |
| --- | --- | --- |
| Выполнил  Студент гр. 951002 |  | В. Н. Протасеня |
| Проверил |  | Асс. Е.Е. Фадеева |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Минск, 2019

1. Теоретические сведения по теме лабораторной работы

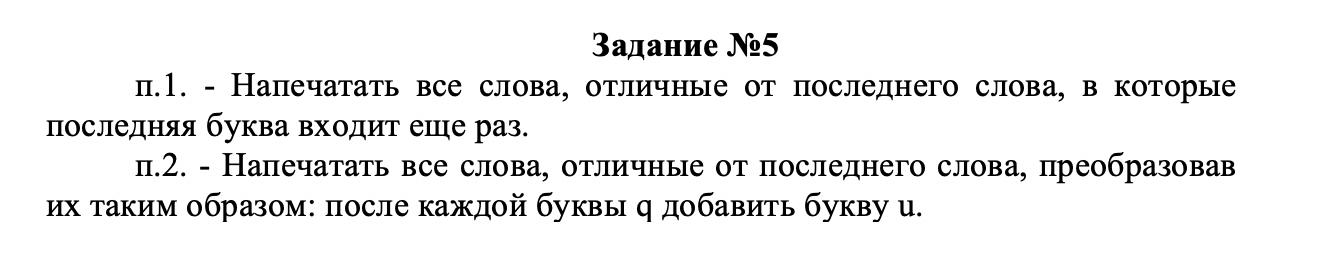
**Строка** – это последовательность любых символов кода обмена информацией (в персональных компьютерах – это код ASCII).

Строка наряду со знаками алфавита языка Паскаль может содержать русские буквы. Строка хранится в поле основной памяти, состоящем из смежных байтов. Каждый байт содержит один символ.

Максимальная длина строки, допускаемая компилятором Паскаль – 255 байт.

Различают строковые константы и строковые переменные.

1.Задание на лабораторную работу



* 1. Эскиз ожидаемого результата

Введиту строку на латинском:

qick licj sos mom dad mom skks sos

строка, отвечающая требованиям первого условия:

mom dad mom skks

строка, отвечающая требованиям второго условия:

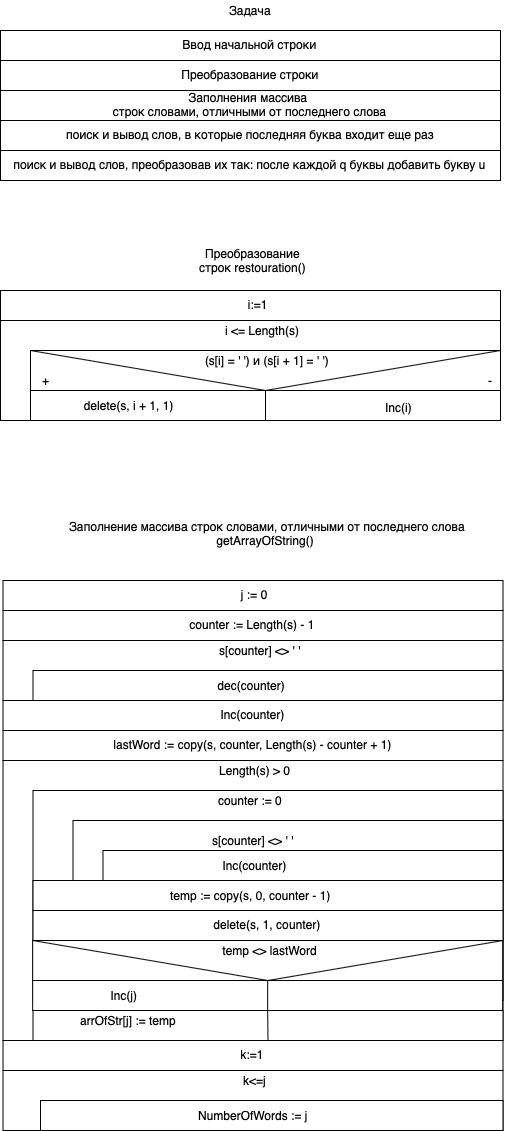
quick licj mom dad mom skks

1. Выполнение
   1. Разработка алгоритма

Таблица 3.1 используемые идентификаторы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя идентификатора | назначение | Тип идентификатора | Начальное значение | Закон изменения | Имя цикла, в котором происходит изменение переменной |
| s | Начальный двумерный массив | Matrix = array [1 .. n, 1 .. m] of integer | — | — | — |
| temp | Одномерный массив для хранения сумм элементов, находящихся по периметру | string | — | res := res + ' ' + IntToStr(perimetr) | 108-128 |
| ArrOfString | Массив строк | Arr = array [1 .. 255] of string | — | arrOfStr[i] := arrOfStr[i] + lastSymbol | procedure getArrayOfString |
| i | Параметр цикла | integer | 1 | Inc(i) | 108-128 |
| lastSymbol | Хранит значение последнего символа в слове | String | — | sum3:=getSumInLine(C, counter) | 70-102 |
| lastWord | Хранит значение последнего слова | String | — | lastWord := copy(s, counter, Length(s) - counter + 1) | 33-70 |
| NumberOfWords | Хранит количество слов в строке | integer | 0 | NumberOfWords := j | 33-70 |
| j | Параметр цикла | integer | 1 | Inc(j) | 36-61 |
| k | Параметр цикла | integer | 1 | Inc(k) | 72 |

Рисунок 3.1 – Схема работы программы методом Насси-Шнейдермана



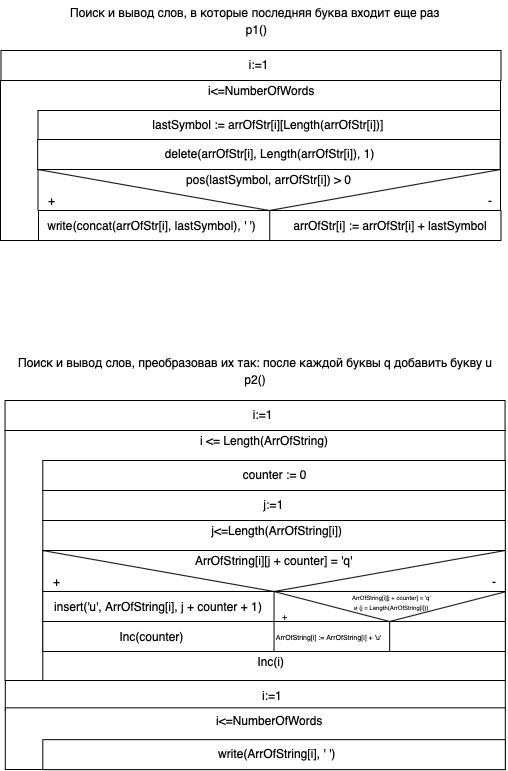
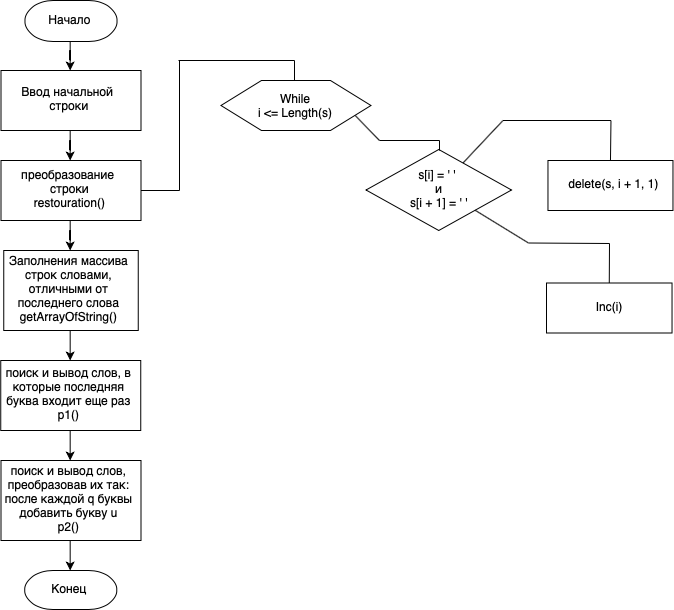
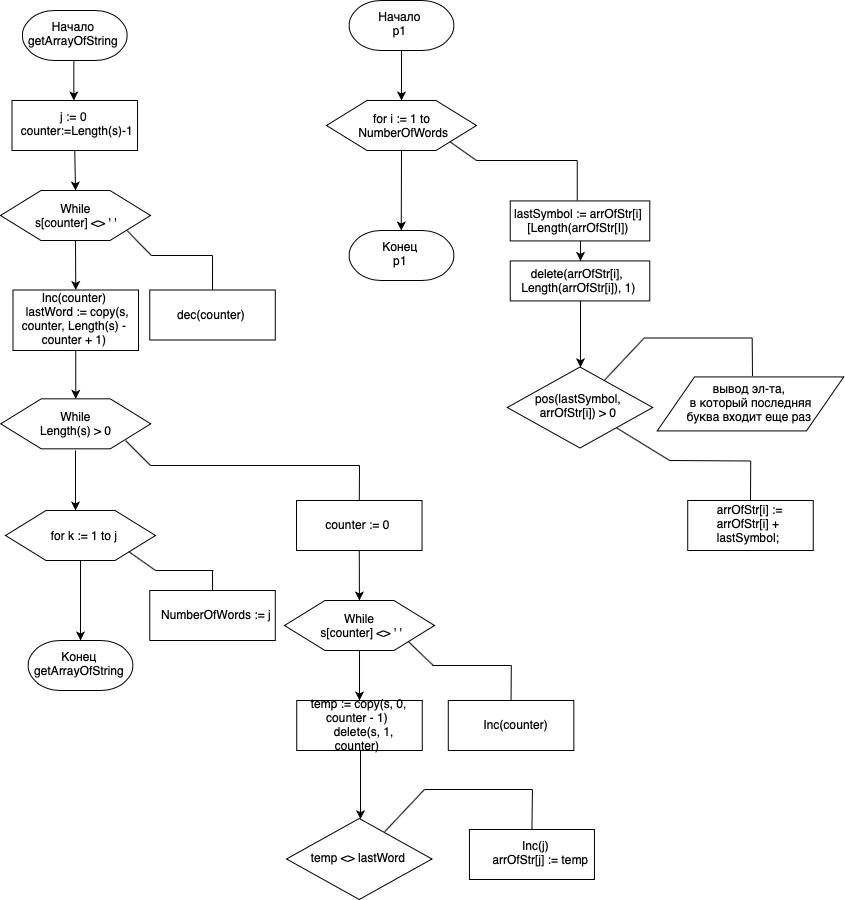
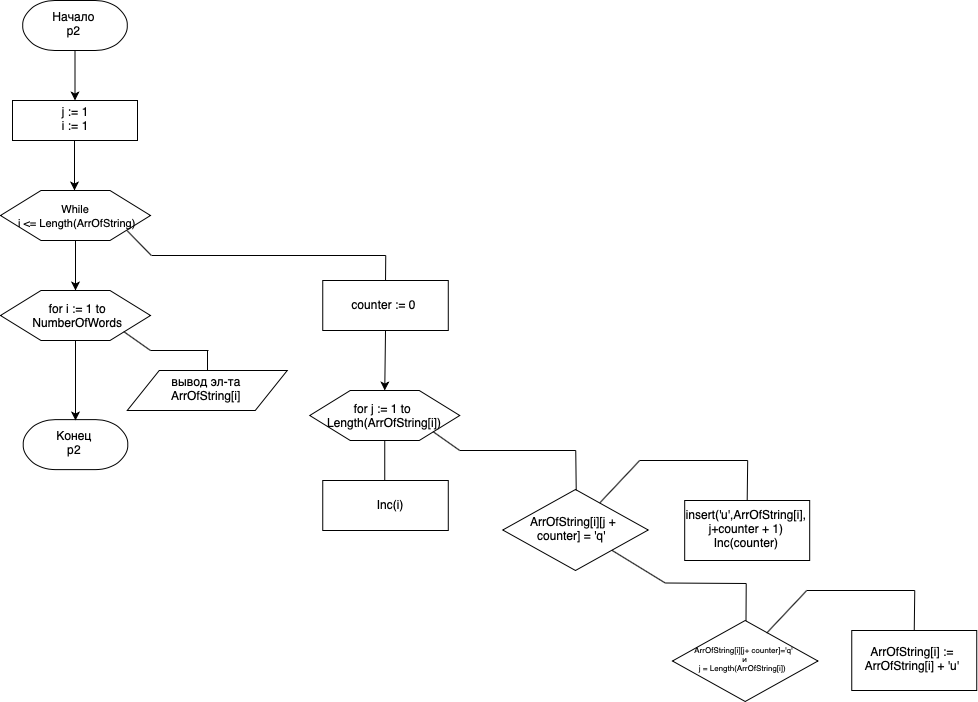


Рисунок 3.2 – Схема работы программы методом Дамке







3.2 Текст программы и его описание

program laba7;

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils, windows;

TYPE

Arr = array [1 .. 255] of string;

var

s, temp: string;

ArrOfString: Arr;

i: integer;

lastWord: string;

j, k: integer;

NumberOfWords: integer;

procedure restouration(var s: string); // удаление лишних пробелов между словами

begin

i := 1;

while i <= Length(s) do //простой с предусловием

begin

if (s[i] = ' ') and (s[i + 1] = ' ') then

delete(s, i + 1, 1)

else

Inc(i);

end;

end;

procedure getArrayOfString(var s: string; var arrOfStr: Arr); // заполнение строкового массива

var

counter, j, k: integer;

temp: string;

lastWord: string;

begin

j := 0;

counter := Length(s) - 1;

while s[counter] <> ' ' do // простой с предусловием

dec(counter);

Inc(counter);

lastWord := copy(s, counter, Length(s) - counter + 1);

while Length(s) > 0 do // сложный внешний с предусловием

begin

counter := 0;

while s[counter] <> ' ' do // простой внутренний с предусловием

Inc(counter);

temp := copy(s, 0, counter - 1);

delete(s, 1, counter);

if temp <> lastWord then

begin

Inc(j);

arrOfStr[j] := temp;

end;

end;

for k := 1 to j do // простой внутренний с предусловием

NumberOfWords := j;

end;

procedure p1(var arrOfStr: Arr); // поиск и вывод слов, в которые последняя буква входит еще раз

var

i: integer;

lastSymbol: string;

word: string;

begin

for i := 1 to NumberOfWords do // простой с предусловием

begin

lastSymbol := arrOfStr[i][Length(arrOfStr[i])]; // поиск последнего символа в слове

delete(arrOfStr[i], Length(arrOfStr[i]), 1); //удаляем

if pos(lastSymbol, arrOfStr[i]) > 0 then //если есть вхождение,то делаем

begin

write(concat(arrOfStr[i], lastSymbol), ' '); //вывод слова с последним символом - конкатинация

arrOfStr[i] := arrOfStr[i] + lastSymbol; // возвращаем последний символ в слово

end

else

arrOfStr[i] := arrOfStr[i] + lastSymbol; // возвращаем последний символ в слово

end;

end;

procedure p2(var ArrOfString: Arr); // добавление после каждой q буквы u

var counter: integer;

begin

j := 1;

i := 1;

while i <= Length(ArrOfString) do //сложный внешний с предусловием

begin

counter := 0;

for j := 1 to Length(ArrOfString[i]) do //простой внутренний с предусловием

begin

if ArrOfString[i][j + counter] = 'q' then

begin

insert('u', ArrOfString[i], j + counter + 1);

Inc(counter);

end

else if (ArrOfString[i][j + counter] = 'q') and

(j = Length(ArrOfString[i])) then

ArrOfString[i] := ArrOfString[i] + 'u';

end;

Inc(i);

end;

for i := 1 to NumberOfWords do //простой с предусловием

write(ArrOfString[i], ' ');

end;

begin

SetConsoleCP(1251);

SetConsoleOutPutCP(1251);

writeln('Введиту строку на латинском:');

readln(s);

restouration(s);

if s = ' ' then

writeln('Строка оказалась пустой, попробуйте еще раз!', s);

restouration(s);

getArrayOfString(s, ArrOfString); //заполнение строкового массива

writeln;

writeln('строка, отвечающая требованиям первого условия:');

p1(ArrOfString);

if ArrOfString[1] = ' ' then

writeln('Строка оказалась пустой, попробуйте еще раз!');

writeln;

writeln;

writeln('строка, отвечающая требованиям второго условия:');

p2(ArrOfString);

if ArrOfString[1] = ' ' then

writeln('Строка оказалась пустой, попробуйте еще раз!');

readln;

end.

* 1. Тестирование и отладка программы

Таблица 3.2 Прохождение тестов программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Специфика тестирования | Номер теста | Вводимые данные | Ожидаемый результат | Полученный результат |
|  | 1 | qqqq ssooso qqq | строка, отвечающая требованиям первого условия:  qqqq ssooso  строка, отвечающая требованиям второго условия:  ququququ ssooso | Тест пройден |
| Введено  1 слово | 2 | qoqo | строка, отвечающая требованиям первого условия:  Строка оказалась пустой, попробуйте еще раз!  строка, отвечающая требованиям второго условия:  Строка оказалась пустой, попробуйте еще раз! | Тест пройден |
| Пустая строка | 3 | Пустая строка | Строка оказалась пустой, попробуйте еще раз! | Тест пройден |
|  | 4 | qwertyuio sos | строка, отвечающая требованиям первого условия:  строка, отвечающая требованиям второго условия:  quwertyuio | Тест пройден |

* 1. Анализ прохождения тестов и полученных результатов

Программа прошла тесты успешно и ошибок не обнаружено.

Итоговый текст программы

program laba7;

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils, windows;

TYPE

Arr = array [1 .. 255] of string;

var

s, temp: string;

ArrOfString: Arr;

i: integer;

lastWord, letter: string;

flag: boolean;

j, k, h: integer;

NumberOfWords: integer;

procedure restouration(var s: string);

begin

i := 1;

while i <= Length(s) do

begin

if (s[i] = ' ') and (s[i + 1] = ' ') then

delete(s, i + 1, 1)

else

Inc(i);

end;

end;

procedure getArrayOfString(var s: string; var arrOfStr: Arr);

var

counter, j, k: integer;

temp: string;

lastWord: string;

begin

j := 0;

counter := Length(s) - 1;

while s[counter] <> ' ' do

dec(counter);

Inc(counter);

lastWord := copy(s, counter, Length(s) - counter + 1);

while Length(s) > 0 do

begin

counter := 0;

while s[counter] <> ' ' do

Inc(counter);

temp := copy(s, 0, counter - 1);

delete(s, 1, counter);

if temp <> lastWord then

begin

Inc(j);

arrOfStr[j] := temp;

end;

end;

for k := 1 to j do

NumberOfWords := j;

end;

procedure p1(var arrOfStr: Arr);

var

i: integer;

lastSymbol: string;

word: string;

begin

for i := 1 to NumberOfWords do

begin

lastSymbol := arrOfStr[i][Length(arrOfStr[i])];

delete(arrOfStr[i], Length(arrOfStr[i]), 1);

if pos(lastSymbol, arrOfStr[i]) > 0 then

write(concat(arrOfStr[i], lastSymbol), ' ')

else

arrOfStr[i] := arrOfStr[i] + lastSymbol;

end;

end;

procedure p2(var ArrOfString: Arr);

var

counter: integer;

begin

j := 1;

i := 1;

while i <= Length(ArrOfString) do

begin

counter := 0;

for j := 1 to Length(ArrOfString[i]) do

begin

if ArrOfString[i][j + counter] = 'q' then

begin

insert('u', ArrOfString[i], j + counter + 1);

Inc(counter);

end

else if (ArrOfString[i][j + counter] = 'q') and

(j = Length(ArrOfString[i])) then

ArrOfString[i] := ArrOfString[i] + 'u';

end;

Inc(i);

end;

for i := 1 to NumberOfWords do

write(ArrOfString[i], ' ');

end;

begin

SetConsoleCP(1251);

SetConsoleOutPutCP(1251);

writeln('Введиту строку на латинском:');

readln(s);

if s = '' then

writeln('Строка оказалась пустой, попробуйте еще раз!', s);

s := trim(s);

restouration(s);

getArrayOfString(s, ArrOfString);

writeln;

writeln('строка, отвечающая требованиям первого условия:');

p1(ArrOfString);

if ArrOfString[1] = '' then

writeln('Строка оказалась пустой, попробуйте еще раз!', ArrOfString[1]);

writeln;

writeln;

writeln('строка, отвечающая требованиям второго условия:');

p2(ArrOfString);

if ArrOfString[1] = '' then

writeln('Строка оказалась пустой, попробуйте еще раз!', ArrOfString[1]);

readln;

end.