БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра программного обеспечения информационных технологий

Факультет ЗВиДО

Специальность ПОИТ

Лабораторная работа № 2

по дисциплине «Основы алгоритмизации и программирования»

часть 1

Вариант № 2

Выполнила: Марфель Т.В.

Минск 2018

1. **Условие задачи**

Дана строка символов S, состоящая из латинских букв. Группы символов, разделенные пробелами и не содержащие пробелов внутри себя, будем называть словами. Преобразовать исходную строку в строки S1 и S2 в соответствии с пунктами 1 и 2 (п.1, п.2) заданий № 1 - 30. Если какая-либо из итоговых строк окажется пустой, выводить соответствующее сообщение.

**Индивидуальное задание №2**

п.1. - Напечатать слово, имеющее минимальную длину и отличное от последнего слова.

п.2. - Напечатать все слова, отличные от последнего слова, удалив все вхождения th, если такие есть.

1. **Листинг программы**

**PROGRAM practice2;**

**Uses Regexpr, Sysutils;**

**Var**

**re : TRegExpr;** {Экземпляр класса TRegExpr}

**ls, s1, s2 : string;** {Исходная и итоговые строки}

**la : array of string;** {Массив для добавления каждого слова строки}

**minLength, i, ind, count : Integer;** Вспомогательные переменные для подсчета минимальной длины и итерации цикла}

**word : boolean;**

**BEGIN**

**ReadLn(ls);** {Чтение данных введенных пользователем}

**ls := Trim(ls**); {Удаление начальных и конечных пробелов во введенной строке при наличии таковых}

**re := TRegExpr.Create;** {Создание экземляра класса TRegExpr для обработки регулярных выражений}

**re.Expression := ('[A-Za-z0-9\_]');** {Словообразующие символы}

**ind := 0;** {Установление индекса для первого элемента массива}

**count := 0;** {Инициализация переменной для подсчета слов в строке}

{Цикл для подсчета слов строке с латинским текстом}

**while i < Length(ls) do**

**begin**

{Если элемент с индексом i не является буквой, значение переменной устанавливается в false и цикл переходит на следующую итерацию}

**if not(re.exec(ls[i])) then**

**word := false**

{Если элемент с индексом i не является буквой и word установлено в false, то счетчик увеличивается на 1, при этом word становится true }

**else if NOT(word) then**

**begin**

**word := true;**

**count := count + 1;**

**end;**

**i := i + 1;**

**end;**

**if (Length(ls) < 1) Or (count = 0) then** {Условие обработки пустой строки и строки, не содержащей словообразующих символов}

**begin**

**WriteLn('Error. The string should not be empty and should contain at least one word character');**

**exit;**

**end;**

**WriteLn('Original string: ');**

**WriteLn(ls, #13#10);** {Вывод изначальной строки}

**WriteLn('Number of words in the string: ', count, #13#10);**

**SetLength(la, count);** {Установление длины динамического массива на основе количества слов во введенной строке}

**word := false;** {Присвоение значения false условной переменной}

{Цикл для заполнения массива словами из введенной строки}

**for i:=0 to Length(ls) + 1 do**

**begin**

**{Если элемент является буквой, то она добавляется в переменную.**

Если подряд встречаются несколько символов, не являющимися буквами, то цикл продолжается.

При совпадении с регулярным выражением слово из переменной сохраняется в элемент массива, а значение s1 удаляется}

**if re.exec(ls[i]) and (i <> Length(ls) + 1) then**

**begin**

**s1 := Concat(s1 + ls[i]);**

**word := true;**

**end**

**else if not(word) then**

**continue**

**else**

**begin**

**word := false;**

**la[ind] := s1;**

**s1 := '';**

**ind := ind + 1;**

**end;**

**end;**

**minLength := Length(la[0]);** {Присвоение значения минимальной длины равной длине нулевого элемента массива}

{Нахождение слова с минимальной длиной в массиве, отличного от последнего слова}

**for i:=0 to Length(la) do**

**begin**

**If (Length(la[i]) < minLength) AND NOT(la[i] = la[Length(la)]) then**

**begin**

**minLength := Length(la[i]);**

**s1 := la[i];**

**end;**

**end;**

**WriteLn('S1 string with a minimum number of chars: ', s1, #13#10);** {Вывод слова с минимальной длиной}

**count := 0;** {Присвоение нулевого значения вспомогательной переменной}

**WriteLn('S2 occurrences');**

{Цикл для нахождения заданной подстроки в каждом элементе массива с ее последующим удаление}

**for i:=0 to Length(la) do**

**begin**

**if pos('th', la[i]) <> 0 then**

**begin**

**s2 := la[i];**

**WriteLn('#', count);**

**Write(s2, #20, '->', #20);**

**Delete(s2, pos('th', s2), 2);**

**Writeln(s2);**

**count := count + 1;**

**end**

{Вывод ошибки при отсутствии искомой подстроки в элементах массива}

**else if (count = 0) and (i = Length(la)) then**

**begin**

**WriteLn('No occurrence of "th" were found. Try again with a different string');**

**end;**

**end;**

**end.**

1. **Тестовый набор данных**
2. **Ввод латинского текста, содержащего th:**

*Входные данные:*

Alienum vivendum ut per. No zril equidem pericula per. In pri suavitate theophrastus, dicam mollis pro ex. Est noster vocent interpretaris no, nostro conclusionemque no mel.

*Результаты работы программы*

|  |
| --- |
| Original string:  Alienum vivendum ut per. No zril equidem pericula per. In pri suavitate theophrastus, dicam mollis pro ex. Est noster vocent interpretaris no, nostro conclusionemque no mel.  Number of words in the string: 26  S1 string with a minimum number of chars: ut  S2 occurrences  #0  theophrastus -> eophrastus |

1. **Ввод пустой строки**

*Входные данные:*

*Результаты работы программы*

|  |
| --- |
| Error. The string should not be empty and should contain at least one word character |

1. **Ввод строки, не содержащей th:**

*Входные данные:*

Sit posse pericula cu, eos ei sint fastidii, ex sint simul eam. An has nullam expetendis, ne impedit feugait scripserit pri, aliquip apeirian mea ad. Ut solet putent dolorem mea, vim ea laudem corrumpit. Ea modo dolores oportere mea, mea graeco gubergren at. Est brute congue recusabo ut, vis detraxit cotidieque eu.

*Результаты работы программы*

|  |
| --- |
| Original string:  Sit posse pericula cu, eos ei sint fastidii, ex sint simul eam. An has nullam expetendis, ne impedit feugait scripserit pri, aliquip apeirian mea ad. Ut solet putent dolorem mea, vim ea laudem corrumpit. Ea modo dolores oportere mea, mea graeco gubergren at. Est brute congue recusabo ut, vis detraxit cotidieque eu.  Number of words in the string: 52  S1 string with a minimum number of chars: cu  S2 occurrences  No occurrence of "th" were found. Try again with a different string |

1. **Ввод строки, состоящей только из пробелов**

*Входные данные:*

*Результаты работы программы*

|  |
| --- |
| Error. The string should not be empty and should contain at least one word character |

1. **Ввод строки, состоящей только из специальных символов**

*Входные данные:*

*,,,,,,,,,,,,,.......(){}[]*

*Результаты работы программы*

|  |
| --- |
| Error. The string should not be empty and should contain at least one word character |