**Министерство образования и наук РФ**

**Федеральное государственное образовательное учреждение**

**Высшего образования**

**Рязанский государственный радиотехнический университет**

**Кафедра ЭВМ**

Отчет по лабораторной работе №7:

“Создание и обработка массивов

символов”

Вариант 12

Выполнила: ст. гр 640

Тограева К. О.

Проверила:

Доц. к.т.н. Елесина С.И.

Рязань 2017

**Цель работы:**  
 Получение навыков обработки информации, представленной в виде массивов символов.

**Вариант №12:**

Подсчитать число слов, у которых левая и правая половины одинаковые.

**Программирование**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include "main.h"**

**//12.**

**//Подсчитать число слов, у которых левая и правая половины**

**//одинаковые.**

**char \*retrieveWord(int length, char \*endPointer);**

**int main() {**

**char \*pWordsArray = (char \*) malloc(1);**

**// массив сохранялка длин слов**

**int lengthArray[20] = {0}; // статический массив - в стеке**

**// указатель на начало массива**

**char \*pArrayBeginning = pWordsArray;**

**int i = 1;**

**printf("\n Введите строку \n");**

**int wordLength = 0;**

**int wordsCounter = 0;**

**char c;**

**do {// todo чтобы дувайл корректно работал**

**c = getchar();**

**if (c != ' ' && c != '\n') {**

**wordLength++;**

**\*pWordsArray = c;**

**pArrayBeginning = (char \*) (realloc(pArrayBeginning, (i += 1) \* sizeof(char)));**

**pWordsArray = pArrayBeginning + i - 1;**

**\*(lengthArray + wordsCounter) = wordLength + 1;**

**} else {**

**wordLength = 0;**

**\*pWordsArray = '\0';**

**// опять выделяем память под следующий символ**

**pArrayBeginning = (char \*) (realloc(pArrayBeginning, (i += 1) \* sizeof(char)));**

**pWordsArray = pArrayBeginning + i - 1;**

**char \*word = retrieveWord(\*(lengthArray + wordsCounter), pWordsArray - 1);**

**printf("%i слово это - %s\n", wordsCounter, word);**

**wordsCounter++;**

**}**

**} while (c != '\n');**

**printf("\n\n\n");**

**printf("Слова, у которых левая и правая половина одинаковые - %i\n",**

**countHalfWords(pArrayBeginning, lengthArray, wordsCounter));**

**return 0;**

**}**

**int countHalfWords(char \*beginning, int lengthArray[20], int wordsCounter) {**

**int mainCounter = 0;**

**for (int i = 0; i < wordsCounter; ++i) {**

**bool founded = true;**

**if (\*(lengthArray + i \* sizeof(char)) % 2 == 0) {**

**founded = false;**

**} else {**

**for (int j = 0; j < (\*(lengthArray + i \* sizeof(char)) / 2); ++j) {// todo это можно сделать через while**

**// если буква слова не равна букве из второй половине, то слово нам не подходит**

**if (\*(beginning + j) != \*(beginning + j + \*(lengthArray + i \* sizeof(char)) / 2)) {**

**founded = false;**

**}**

**}**

**}**

**if (founded) {**

**mainCounter++;**

**char \*word = retrieveWord(\*(lengthArray + i),**

**beginning + \*(lengthArray + i) - 1);// todo добавить перегр с указателем на начало**

**printf("Нашли слово с одинаковой левой и правой половиной - %s\n", word);**

**}**

**// переходим к началу следующего элемента**

**beginning = beginning + \*(lengthArray + i);**

**}**

**return mainCounter;**

**}**

**char \*retrieveWord(int length, char \*endPointer) {**

**char \*word = nullptr;**

**int letter = 0;**

**word = (char \*) malloc(1);**

**char \*startPointer = endPointer - length + 1;**

**for (; startPointer < endPointer; startPointer++) {**

**\*(word + letter) = \*startPointer;**

**letter++;**

**word = (char \*) realloc(word, (letter + 1) \* sizeof(char));**

**}**

**\*(word + letter) = '\0';**

**return word;**

**}**

Тестовый пример:

Пример 1

wowo he climed to me

0 слово это - wowo

1 слово это - he

2 слово это - climed

3 слово это - to

4 слово это - me

Нашли слово с одинаковой левой и правой половиной - wowo

Слова, у которых левая и правая половина одинаковые – 1

Пример 2

so give me a reason to prove me wrong

0 слово это - so

1 слово это - give

2 слово это - me

3 слово это - a

4 слово это - reason

5 слово это - to

6 слово это - prove

7 слово это - me

8 слово это - wrong

Слова, у которых левая и правая половина одинаковые - 0

Вывод:

Мы научились работать информацией, представленной в виде массивов символов.