

Министерство образования Республики Беларусь

Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Отчёт  
по лабораторной работе № 7  
«Функции»

по курсу:  
«Основы алгоритмизации и программирования»

Выполнил:  
Студент группы 950503  
Полховский А.Ф.

Проверил:  
Дубовский А.Л.

Минск 2019

Цель работы: Научиться создавать и использовать новые функции пользователя.

Вариант 4

Условие:

10) Дан текст, состоящих из n предложений. Слова в предложениях разделены одним или несколькими пробелами. В головной функции ввести текст. Во второй функции записать без повторения в новый массив слова, встречающиеся в тексте более одного раза.

Код программы на языке Си:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>

int resultCount = 0;

int compareWords(char *word1, char *word2) {
    int i = 0;
    int isEqual = 1;
    while (word1[i] != '\0' && word2[i] != '\0') {
        if (word1[i] != word2[i]) {
            isEqual = 0;
            break;
        }
        i++;
    }
    if (word1[i] != word2[i]) isEqual = 0;
    return isEqual;
}

char* parse(char *text) {
    int i = 0;
    int wordLength = 0;
    int wordsCount = 0;
    char **words = (char**)malloc((1) * sizeof(char*)); // выделение памяти
    while (text[i] != '\0') {
        if (text[i] != ' ' && text[i] != '.' && text[i] != ',') {
            if (wordLength == 0) {
                words[wordCount] = (char *)calloc(2, sizeof(char)); //2 char
                //на 1 символ и на '\0'
            }
            else {
                words[wordCount] = (char *)realloc(words[wordCount],
                (wordLength + 1));
            }
            words[wordCount][wordLength] = text[i];
            wordLength++;
        }
        if (text[i] == ' ' || text[i] == '.' || text[i] == ',' || text[i + 1] ==
        '\0') { // i+1 для while(не сработает посл раз)
            if (wordLength) { //если слово есть
                words[wordCount][wordLength] = '\0';
                wordsCount++; // увеличение количества слов
                words = (char**) realloc(words, (wordsCount + 1) * sizeof(char
                *));
                wordLength = 0;
            }
        }
        i++;
    }
}
```

```

    }

    for (int i = 0; i < wordsCount; i++) {
        printf("\nword = %s", words[i]);
    }

    char **result = (char**)malloc((wordsCount / 2) * sizeof(char*));
    for (int i = 0; i < wordsCount; i++) {
        for (int j = i + 1; j < wordsCount; j++) {
            if (compareWords(words[i], words[j]) == 1) {
                int wasBefore = 0;
                for (int k = 0; k < i; k++) {
                    if (compareWords(words[k], words[i])) {
                        wasBefore = 1;
                        break;
                    }
                }
                if (wasBefore == 0) {
                    result[resultCount] = words[i]; // можно делать realloc
                    resultCount++;
                }
                break;
            }
        }
    }
    return result;
}

int main() {
    setlocale(LC_ALL, "RUSSIAN");
    int size = 10, i = 0;
    printf("Input text\n");
    char *text = (char *)calloc(size, sizeof(char));
    rewind(stdin);
    do {
        text[i] = getchar();
        if (text[i] == '\n') break;
        i++;
        if (i >= size) {
            size *= 2;
            text = (char *)realloc(text, size * sizeof(char));
        }
    } while (text[i - 1] != ' ' || text[i - 1] != EOF);
    text[i] = '\0';
    //char* text = " ,hello  hello hi. eh hi eh. ld, ld , hi\0";
    printf("%s", text);

    char **result = (char**)malloc((10) * sizeof(char*));

    result = parse(text);

    for (int i = 0; i < resultCount; i++) {
        printf("\n%s", result[i]);
    }
    getchar();
    return 0;
}

```

Вывод: ознакомился с основными теоритическими сведениями; а также разработал, написал и отладил программу 4-ого варианта.