Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Отчёт

по лабораторной работе № 6

«Символьные строки.

Массивы символьных строк»

по курсу:

«Основы алгоритмизации и программирования»

Выполнил: Проверил:

Студент группы 950503

Гуринович Андрей Викторович Дубовский Андрей Леонидович

Минск 2019

Цель работы: освоить методику работы с символьными строками. Научиться использовать функции для работы со строками.

Научиться объявлять и использовать массивы строк и функции для работы со строками.

Вариант: 5

Задание:  
 1) Даны натуральное число n и символы s1,...,sn. Известно, что среди данных символов есть хотя бы один, отличный от пробела. Требуется преобразовать последовательность s1,...,sn следующим образом. Удалить группы пробелов, которыми начинается и которыми заканчивается последовательность, а также заменить каждую внутреннюю группу пробелов одним пробелом. Если указанных групп нет в данной последовательности, то оставить последовательность без изменения.

2) Дан массив строк. Выделить слова из строк и записать их в массив. Необходимо рассортировать массив слов по их длинам.

Код программы:

1)

#include <stdio.h>

#include <stdlib.h>

**char**\* createAndScanString(**int** stringLenght){

**char** \*string;

**char** error = 0;

**do**{

**if** (!(string = (**char** \*)malloc(stringLenght))){

error = 1;

}

}**while** (error == 1);

**do**{

fflush(stdin);

printf("String: ");

gets(string);

}**while** (string[stringLenght] != '\0');

**return** string;

}

**int** main() {

**int** stringLenght;

**int** withError;

**do**{

printf("String lenght: ");

rewind(stdin);

withError = scanf("%d", &stringLenght);

}**while** (!withError || stringLenght < 1);

**char**\* string = createAndScanString(stringLenght);

**int** i = 0;

**while** (string[i] == ' ') {

stringLenght--;

i++;

}

**for**(**int** j = 0; j < stringLenght; j++){

string[j] = string[j + i];

}

i = stringLenght - 1;

**while** (string[i] == ' ') {

i--;

stringLenght--;

}

string[stringLenght] = '\0';

i = 1;

**int** temp;

**while**(i < stringLenght - 1){

**if**(string[i] == ' ' && string[i + 1] == ' '){

temp = 0;

**while** (string[i + 1 + temp] == ' ') {

temp++;

}

stringLenght -= temp;

**for** (**int** j = i; j < stringLenght; j++){

string[j] = string[j + temp];

}

}

i++;

}

string[stringLenght] = '\0';

string = (**char**\* )realloc(string, stringLenght);

printf("Result:%s\n", string);

free(string);

**return** 0;

}

2)

#include <stdio.h>

#include <stdlib.h>

**char**\* createAndScanString(){

**char** \*string;

**int** stringLenght;

**do**{

printf("String lenght: ");

rewind(stdin);

scanf("%d", &stringLenght);

}**while** (stringLenght < 1);

**char** error = 0;

**do**{

**if** (!(string = (**char** \*)malloc(stringLenght))){

error = 1;

}

}**while** (error == 1);

**do**{

fflush(stdin);

printf("String: ");

gets(string);

}**while** (string[stringLenght] != '\0');

**return** string;

}

**int** main() {

**int** strngsNumber;

**do**{

printf("strngsNumber: ");

rewind(stdin);

}**while**(!scanf("%d", &strngsNumber) || strngsNumber < 1);

**char**\*\* strings = (**char**\*\*)malloc(**sizeof**(**char**\*) \* strngsNumber);

**char**\* string = **NULL**;

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

strings[i] = createAndScanString();

}

**int** i;

**int** firstWordIndex = 0;

**int** wordLenght = 0;

**int** wordNumber = 0;

**char** \*\*words = (**char** \*\*)malloc(wordNumber \* **sizeof**(**char**\*));

**int** \*wordLenghts = (**int** \*)malloc(wordNumber \* **sizeof**(**int**));

**char** \*word = **NULL**;

**for** (strngsNumber -= 1; strngsNumber >= 0; strngsNumber--){

string = strings[strngsNumber];

i = 0;

**while**(string[i] != '\0'){

wordLenght = 0;

**while**(string[i] == ' '){

i++;

}

**if**(string[i] != '\0'){

firstWordIndex = i;

}

**while**(string[i] != ' ' && string[i] != '\0'){

wordLenght++;

i++;

}

**if**(wordLenght){

word = (**char**\*)malloc(wordLenght \* **sizeof**(**char**));

**for**(**int** j = 0; j < wordLenght; j++){

word[j] = string[firstWordIndex + j];

}

word[wordLenght] = '\0';

wordNumber++;

words = (**char**\*\*)realloc(words, **sizeof**(**char**\*) \* wordNumber);

words[wordNumber - 1] = word;

wordLenghts = (**int** \*)realloc(wordLenghts, wordNumber \* **sizeof**(**int**));

wordLenghts[wordNumber - 1] = wordLenght;

}

}

}

free(string);

free(strings);

**int** tempLength;

**int** minimalIndex;

**for**(i = 0; i < wordNumber - 1; i++){

minimalIndex = i;

**for**(**int** j = i; j < wordNumber; j++){

**if**(wordLenghts[minimalIndex] > wordLenghts[j]){

minimalIndex = j;

}

}

**if** (minimalIndex == i){

**continue**;

}

word = words[minimalIndex];

words[minimalIndex] = words[i];

words[i] = word;

tempLength = wordLenghts[minimalIndex];

wordLenghts[minimalIndex] = wordLenghts[i];

wordLenghts[i] = tempLength;

}

free(wordLenghts);

free(word);

**for**(**int** j = 0; j < wordNumber; j++){

printf("\n%s",words[j]);

}

free(words);

**return** 0;

}

Вывод:

1) Создана типовая функция для создания символьной строки, изучены методы удаления лишних пробелов в строке (форматирование строки).

2) Изучен процесс обработки указателей и указателей на указатель по отношению к символьным строкам, обнаружена важность установки символа ‘\0’ при заполнении строки не использую scanf.