Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Отчёт

по лабораторной работе № 8

«Указатели на функцию. Рекурсия.»

по курсу:

«Основы алгоритмизации и программирования»

Выполнил: Проверил:

Студент группы 950503

Гуринович Андрей Викторович Дубовский Андрей Леонидович

Минск 2019

Цель работы: научиться использовать в программах указатели на функции

Вариант: 5

Задание: Дан массив строк. Необходимо сформировать массив указателей на первые по алфавиту n слов из этих предложений. Память под массивы отводить динамически. Использовать указатели на функции.

Код программы:

#include <stdio.h>

#include <stdlib.h>

**int** scanNumberWithLabel(**char**\* string){

**int** number;

**do**{

printf("%s", string);

rewind(stdin);

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

**return** number;

}

**char**\* createAndScanString(){

**int** stringLenght = scanNumberWithLabel("Lenght of string: ");

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

**do**{

fflush(stdin);

printf("String: ");

gets(string);

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

**return** string;

}

**int** main(){

**int** (\*scanNumber)(**char**\* string) = scanNumberWithLabel;

**char**\* (\*createString)(**void**) = createAndScanString;

**int** stringsNumber = (\*scanNumber)("Number of strings: ");

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

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

strings[i] = (\*createString)();

}

**int** i;

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

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

**int** firstWordIndex;

**int** wordLenght;

**int** wordsNumber = 0;

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

**for**(**int** temp = 0; temp < stringsNumber; temp++){

string = strings[temp];

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 && (string[firstWordIndex] >= 'A' && string[firstWordIndex] <= 'Z') || (string[firstWordIndex] >= 'a' && string[firstWordIndex] <= 'z')){

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

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

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

}

word[wordLenght] = '\0';

wordsNumber++;

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

words[wordsNumber - 1] = word;

}

}

}

free(string);

free(strings);

**if**(!wordsNumber){

printf("No words");

**return** 0;

}

**int** tempNumber;

**int** numberOfSortingWords;

**do**{

numberOfSortingWords = (\*scanNumber)("Number of words to sort: ");

} **while** (numberOfSortingWords > wordsNumber || numberOfSortingWords < 1);

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

tempNumber = i;

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

**if**(words[tempNumber][0] > words[j][0]){

tempNumber = j;

}

}

**if**(tempNumber != i){

word = words[tempNumber];

words[tempNumber] = words[i];

words[i] = word;

}

}

free(word);

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

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

puts(words[i]);

}

free(words);

**return** 0;

}

Вывод: использование указателей на функцию может быть удобно при циклических процессах, к примеру, пользователь должен выбрать тип операции, которая будет проводиться с данными, без указателей пришлось бы делать switch для выполнения каждой такой операций, если они используются в коде многократно, а с указателями switch будет только один: он будет задавать указателю значение необходимой функции, что упростит код.