# ЦЕЛЬ РАБОТЫ

Изучение принципов организации и программирование ввода-вывода в языке С.

# ПОСТАНОВКА ЗАДАЧИ

* В соответствии с вариантом №7: разработать программу поиска самого длинного слова в строке. Считать, что строка оканчивается точкой, слова разделены пробелами.
* В программе предусмотреть ввод и вывод исходных данных и результатов

# СТРУКТУРНЫЕ СХЕМЫ АЛГОРИТМОВ



Рисунок 1 – Схема алгоритма функции Interface



Рисунок 2 – Схема алгоритма функции проверки открытия файла CheckOpen



Рисунок 3 – Схема алгоритма функции поиска самого длинного слова в файле SearchLongestWord

# ТЕКСТ ПРОГРАММЫ

#include <stdio.h>  
#include <stdlib.h>  
#include <ctype.h>  
#include <string.h>  
const unsigned char FILE\_NAME\_MAX\_SIZE = 256;  
void Interface();  
void CheckOpen(FILE \* fileName);  
char \*SearchLongestWord(FILE \* fileName);  
int main() {  
 Interface();  
 system("pause");  
 return 0;}

void Interface(){  
 char outFileName[FILE\_NAME\_MAX\_SIZE];  
 char inFileName[FILE\_NAME\_MAX\_SIZE];  
 //READ FILE  
 printf("ENTER READ FILE NAME: ");  
 gets(outFileName);  
 FILE \*outFile = fopen(outFileName, "r");  
 CheckOpen(outFile);  
 putchar('\n');  
 //WRITE TO FILE  
 printf("ENTER WRITE FILE NAME: ");  
 gets(inFileName);  
 FILE \*inFile = fopen(inFileName, "w");  
 CheckOpen(inFile);  
 char \*longestWord = SearchLongestWord(outFile);  
 fputs(longestWord,inFile);  
 printf("%d - %s \n",strlen(longestWord), longestWord);  
 free(longestWord);  
 fclose(outFile);  
 fclose(inFile);  
}  
  
void CheckOpen(FILE \* file){  
 if (!file){  
 printf("OPEN FILE ERROR\n");

fclose(file);  
 system("pause");  
 exit(1);  
 } else { printf("FILE WAS OPENED\n");  
 }}

char \*SearchLongestWord(FILE \* file){  
 rewind(file);  
 char tempChar = 0; size\_t charCounter = 0; size\_t maxLength = 0;  
 size\_t begWordPosition = 0; size\_t longestWordPos = 0;  
 while ( fscanf(file, "%c", &tempChar)!= EOF){  
 if (!ispunct(tempChar) && !isspace(tempChar)) {  
 if (!charCounter) begWordPosition = ftell(file);

charCounter++; } else {  
 if (charCounter > maxLength) {  
 maxLength = charCounter;  
 longestWordPos = begWordPosition - 1;  
 }charCounter = 0;  
 } }  
 char \*word = (char\*) calloc (maxLength + 1,sizeof(char));  
 fseek(file, longestWordPos, SEEK\_SET);  
 fgets(word, maxLength+1, file);   
 return word;}

# ТЕСТИРОВАНИЕ ПРОГРАММЫ

Для проверки работоспособности программы было разработано несколько тестовых примеров:

* Ввод имени несуществующего исходного файла:

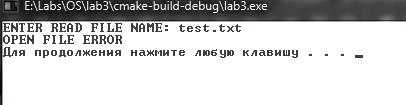


Рисунок 4 – Результат выполнения программы при попытке открыть несуществующий файл ввода

* Ввод имени несуществующего файла вывода:

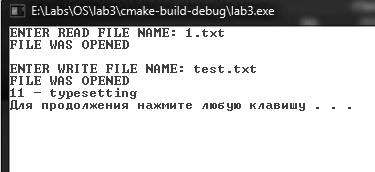


Рисунок 5 – Результат выполнения программы при попытке открыть несуществующий файл вывода

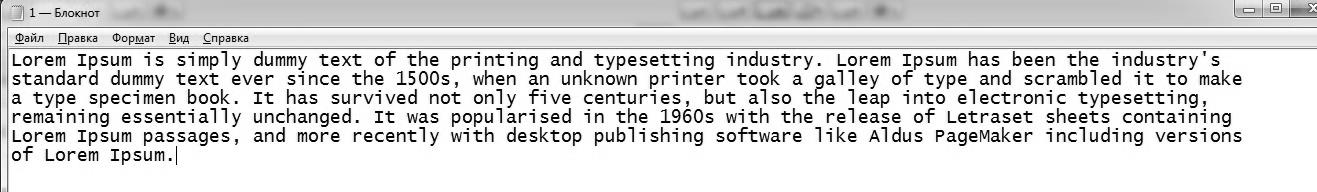


Рисунок 6 – Содержимое исходного файла

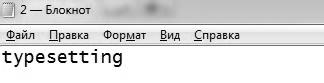


Рисунок 7 – Содержимое файла вывода

ВЫВОД

При выполнении лабораторной работы была написана программа на языке C, которая позволяет открыть файл, запрашиваемый пользователем и произвести поиск самого длинного слова в этом файле. Результат работы программы выводится на экран, а так же в другой файл выбранный пользователем,.

По результатам тестирования, можно сделать вывод, что программа работоспособна и поставленная задача достигнута.