Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Кафедра ІПІ

**ЗВІТ**

з виконання лабораторної роботи № 1

з кредитного модуля

“Основи програмування-2. Методології програмування”

**Варіант № 5**

Виконав:

Студент 1-го курсу

Групи ІП-22 ФІОТ

Гринько Олександр Максимович

Київ 2023

**Постановка задачі**

Изображение выглядит как текст

Автоматически созданное описание

**Вимоги до програми**

Изображение выглядит как текст

Автоматически созданное описание

**Код програми**

**main.cpp**

#include "FunctionsList.h"

#include "FunctionsListC.h"

int main(int argc, char\* argv[]) {

char mode[] = "mode";

char mode1[] = "FP";

char mode2[] = "FS";

if (argc != 3 || strcmp(argv[1], mode) != 0) {

cout << "Launch the program with particular mode!\n" <<

"exeName.exe mode FP/FS\n";

}

else if (!strcmp(argv[2], mode1)) {

namesC();

}

else if (!strcmp(argv[2], mode2)) {

names();

}

else {

cout << "The only modes are FP/FS";

}

system("pause");

}

**Functions.cpp**

#include "FunctionsList.h"

void names() {

string nameIn = "input.txt";

string nameOut = "output.txt";

string nameF = "file.txt";

inputFile(nameIn, nameOut, nameF);

}

void inputFile(string name, string name1, string name2) {

ofstream nameIn;

nameIn.open(name);

string s;

int m;

int stop\_symbol;

cout << "Enter the text:" << endl;

do {

getline(cin, s);

stop\_symbol = s.find('&');

if (stop\_symbol == string::npos) nameIn << s << "\n";

else nameIn << s;

} while (stop\_symbol == string::npos);

nameIn.close();

deleteLastSymbol(name, name1, name2);

cout << "Do you want to add something?\n1 - yes\n2 - no" << endl;

cin >> m;

if (m == 1) addNewText(name, name1, name2);

else if (m == 2) makeArrayOfWords(name1, name2);

textOut(name, name1);

}

void addNewText(string name, string name1, string name2) {

fstream nameIn;

nameIn.open(name, fstream::app);

string s;

int m;

int stop\_symbol;

cout << "Enter the text:" << endl;

do {

getline(cin, s);

stop\_symbol = s.find('&');

if (stop\_symbol == string::npos) nameIn << s << "\n";

else nameIn << s;

} while (stop\_symbol == string::npos);

nameIn.close();

deleteLastSymbol(name, name1, name2);

cout << "Do you want to add something?\n1 - yes\n2 - no" << endl;

cin >> m;

if (m == 1) addNewText(name, name1, name2);

else if (m == 2) makeArrayOfWords(name1, name2);

}

void deleteLastSymbol(string name, string name1, string name2) {

ifstream nameIn;

nameIn.open(name);

ofstream nameOut;

nameOut.open(name1);

string s;

int position;

int lengthstring;

while (!nameIn.eof()) {

getline(nameIn, s);

lengthstring = s.length();

position = s.find('&');

if (position >= 0) s.erase(position, lengthstring - position);

if (!nameIn.eof()) nameOut << s << endl;

else nameOut << s;

}

nameOut.close();

nameIn.close();

remove(name.c\_str());

rename(name1.c\_str(), name.c\_str());

deleteCommasAndDots(name, name1, name2);

}

void deleteCommasAndDots(string name, string name1, string name2) {

ifstream nameIn;

nameIn.open(name);

ofstream nameOut;

nameOut.open(name1);

string s;

int pos;

int i = 0;

while (!nameIn.eof()) {

getline(nameIn, s, ' ');

if ((s.find(',') != string::npos) || (s.find('.') != string::npos))

while ((s.find(',') != string::npos) || (s.find('.') != string::npos)) {

if (s.find(',') != string::npos) pos = s.find(',');

else if (s.find('.') != string::npos) pos = s.find('.');

s.erase(pos, 1);

}

nameOut << s << '\n';

}

nameIn.close();

nameOut.close();

}

void makeArrayOfWords(string name1, string name2) {

ifstream nameOut;

nameOut.open(name1);

string s;

int i = 0;

string words[300];

while (!nameOut.eof()) {

getline(nameOut, s);

if (s != "") {

words[i] = s;

i++;

}

}

nameOut.close();

moreThanN(name1, name2, words, i);

}

void moreThanN(string name1, string name2, string arr[], int size) {

int N;

cout << "Enter the N: ";

cin >> N;

ofstream nameF;

nameF.open(name2);

ofstream nameOut;

nameOut.open(name1);

int k = 0;

int index = 0;

string s;

string repeatWords[300];

int indexArray[300];

for (int i = 0; i < size; i++) {

bool f = true;

int count = 1;

for (int j = i + 1; j < size; j++)

if (arr[i] == arr[j]) {

indexArray[index] = j;

index++;

count++;

}

if (count > N) {

for (int v = 0; v <= index; v++)

if (indexArray[v] == i) {

f = false;

break;

}

if (f) {

repeatWords[k] = arr[i];

k++;

}

}

}

for (int i = 0; i < k - 1; i++)

for (int j = 0; j < k - i - 1; j++)

if (repeatWords[j].length() > repeatWords[j + 1].length()) {

swap(repeatWords[j], repeatWords[j + 1]);

}

for (int i = 0; i < k; i++)

nameF << repeatWords[i] << '\n';

nameF.close();

nameOut.close();

remove(name1.c\_str());

rename(name2.c\_str(), name1.c\_str());

}

void textOut(string name, string name1) {

ifstream nameOut;

nameOut.open(name);

ifstream nameOut1;

nameOut1.open(name1);

string s;

cout << "\n\n\n----------------- Our file1 -----------------\n\n\n";

while (!nameOut.eof()) {

getline(nameOut, s);

cout << s << endl;

}

cout << "\n\n\n----------------- Our file2 -----------------\n\n\n";

while (!nameOut1.eof()) {

getline(nameOut1, s);

cout << s << endl;

}

}

**FunctionsC.cpp**

#include "FunctionsListC.h"

void namesC() {

char nameIn[] = "input.txt";

char nameOut[] = "output.txt";

char nameF[] = "file.txt";

inputFileC(nameIn, nameOut, nameF);

}

void inputFileC(char\* name, char\* name1, char\* name2) {

FILE\* nameIn = fopen(name, "w");

char s[1000];

int stop\_symbol;

int m;

printf("Enter the text:\n");

do {

fgets(s, 1000, stdin);

stop\_symbol = 0;

for (int i = 0; s[i] != '\0'; i++) {

if (s[i] == '&') {

stop\_symbol = 1;

break;

}

}

if (!stop\_symbol) fprintf(nameIn, "%s", s);

else fprintf(nameIn, "%s", s);

} while (!stop\_symbol);

fclose(nameIn);

deleteLastSymbolC(name, name1, name2);

printf("Do you want to add something?\n1 - yes\n2 - no\n");

scanf("%d", &m);

if (m == 1) addNewTextC(name, name1, name2);

else if (m == 2) makeArrayOfWordsC(name1, name2);

textOutC(name, name1);

}

void addNewTextC(char\* name, char\* name1, char\* name2) {

FILE\* nameIn = fopen(name, "a");

char s[100];

int stop\_symbol;

printf("Enter the text:\n");

int m;

do {

fgets(s, 100, stdin);

stop\_symbol = -1;

for (int i = 0; s[i] != '\0'; i++) {

if (s[i] == '&') {

stop\_symbol = i;

break;

}

}

if (stop\_symbol == -1) fprintf(nameIn, "%s", s);

else fprintf(nameIn, "%s", s);

} while (stop\_symbol == -1);

fclose(nameIn);

deleteLastSymbolC(name, name1, name2);

printf("Do you want to add something?\n1 - yes\n2 - no\n");

scanf("%d", &m);

if (m == 1) addNewTextC(name, name1, name2);

else if (m == 2) makeArrayOfWordsC(name1, name2);

}

void deleteLastSymbolC(char\* name, char\* name1, char\* name2) {

FILE\* nameIn = fopen(name, "r");

FILE\* nameOut = fopen(name1, "w");

char s[1000];

int position;

int lengthstring;

while (fgets(s, 1000, nameIn)) {

lengthstring = 0;

while (s[lengthstring] != '\0') {

lengthstring++;

}

position = -1;

for (int i = 0; i < lengthstring; i++) {

if (s[i] == '&') {

position = i;

break;

}

}

if (position >= 0) {

s[position] = '\0';

}

fprintf(nameOut, "%s", s);

}

fclose(nameOut);

fclose(nameIn);

remove(name);

rename(name1, name);

deleteCommasAndDotsC(name, name1, name2);

}

void deleteCommasAndDotsC(char\* name, char\* name1, char\* name2) {

FILE\* nameIn = fopen(name, "r");

FILE\* nameOut = fopen(name1, "w");

char s[1000];

char ch;

int pos = 0;

while ((ch = fgetc(nameIn)) != EOF) {

if (isspace(ch) || (ispunct(ch) && ch != '\'')) {

if (pos > 0) {

s[pos] = '\0';

fprintf(nameOut, "%s\n", s);

pos = 0;

}

}

else {

s[pos++] = ch;

}

}

if (pos > 0) {

s[pos] = '\0';

fprintf(nameOut, "%s\n", s);

}

fclose(nameIn);

fclose(nameOut);

}

void makeArrayOfWordsC(char\* name1, char\* name2) {

FILE\* nameOut;

nameOut = fopen(name1, "r");

char s[100];

int i = 0;

char\*\* words = (char\*\*)malloc(sizeof(char\*) \* 300);

for (int j = 0; j < 300; j++) {

words[j] = (char\*)malloc(sizeof(char) \* 100);

}

while (fgets(s, sizeof(s), nameOut)) {

if (s[0] != '\n') {

sscanf(s, "%s", words[i]);

i++;

}

}

fclose(nameOut);

moreThanNC(name1, name2, words, i);

}

void moreThanNC(char\* name1, char\* name2, char\*\* arr, int size) {

int N;

printf("Enter the N: ");

scanf("%d", &N);

FILE\* nameF;

nameF = fopen(name2, "w");

FILE\* nameOut;

nameOut = fopen(name1, "w");

int k = 0;

int index = 0;

char s[300];

char\* repeatWords[300];

int indexArray[300];

for (int i = 0; i < size; i++) {

bool f = 1;

int count = 1;

for (int j = i + 1; j < size; j++)

if (strcmp(arr[i], arr[j]) == 0) {

indexArray[index] = j;

index++;

count++;

}

if (count > N) {

for (int v = 0; v <= index; v++)

if (indexArray[v] == i) {

f = 0;

break;

}

if (f) {

repeatWords[k] = arr[i];

k++;

}

}

}

for (int i = 0; i < k - 1; i++)

for (int j = 0; j < k - i - 1; j++)

if (strlen(repeatWords[j]) > strlen(repeatWords[j + 1])) {

char\* temp = repeatWords[j];

repeatWords[j] = repeatWords[j + 1];

repeatWords[j + 1] = temp;

}

for (int i = 0; i < k; i++)

fprintf(nameF, "%s\n", repeatWords[i]);

fclose(nameF);

fclose(nameOut);

remove(name1);

rename(name2, name1);

}

void textOutC(char\* name, char\* name1) {

FILE\* nameOut;

nameOut = fopen(name, "r");

FILE\* nameOut1;

nameOut1 = fopen(name1, "r");

char s[100];

printf("\n\n\n----------------- Our file1 -----------------\n\n\n");

while (fgets(s, sizeof(s), nameOut)) {

printf("%s", s);

}

printf("\n\n\n----------------- Our file2 -----------------\n\n\n");

while (fgets(s, sizeof(s), nameOut1)) {

printf("%s", s);

}

fclose(nameOut);

fclose(nameOut1);

}

**FunctionsList.h**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

void names();

void inputFile(string, string, string);

void deleteLastSymbol(string, string, string);

void addNewText(string, string, string);

void deleteCommasAndDots(string, string, string);

void moreThanN(string, string, string arr[], int size);

void makeArrayOfWords(string, string);

void textOut(string, string);

**FunctionsListC.h**

#include <stdio.h>

#include <cstdlib>

#include <stdlib.h>

#include <ctype.h>

#include <string.h>

void namesC();

void inputFileC(char\*, char\*, char\*);

void deleteLastSymbolC(char\*, char\*, char\*);

void addNewTextC(char\*, char\*, char\*);

void deleteCommasAndDotsC(char\*, char\*, char\*);

void moreThanNC(char\*, char\*, char\*\* arr, int size);

void makeArrayOfWordsC(char\*, char\*);

void textOutC(char\*, char\*);

**Результат роботи**

**mode FS**

Изображение выглядит как текст, снимок экрана, в помещении, черный

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

**mode FP**

Изображение выглядит как текст, снимок экрана, в помещении, черный

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание