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 № 26

Виконав:

студент 1-го курсу

гр. ІП-21 ФІОТ

Скрипець Ольга Олександрівна

Київ 2023

1. **Посилання на репозиторій**

https://github.com/uhpolerr/op\_lab1

1. **Код програми**

**Files.h**

#pragma once

#include <iostream>

#include <fstream>

#include <string>

#include <conio.h>

#include <cstdio>

#include <algorithm>

#include <regex>

#include <sstream>

using namespace std;

string get\_filename();

char\* get\_filename1();

void create\_file(const string& filename);

void create\_file(char filename[]);

void text\_to\_file(string& filename);

void text\_to\_file(char\* filename);

void open\_file(string& filename);

void open\_file(char\* filename);

int enter\_k();

string reverse\_string(string line);

string encrypt\_string(string line, int k);

char\* reverse\_string(char\* line, int k);

char\* encrypt\_string(char\* line, int k, int n);

void encryption(string& filename, int k);

void encryption(char\* filename, int k);

void choose\_line();

void choose\_line1();

bool is\_word\_latin(char\* word, int n);

bool is\_word\_latin(string word);

void remove\_latin\_words();

void remove\_latin\_words1();

**Files.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <fstream>

#include <string>

#include <conio.h>

#include <cstdio>

#include <algorithm>

#include <regex>

#include <sstream>

#include "Files.h"

using namespace std;

//функція для запиту в користувача імені файлу

string get\_filename() {

string filename;

bool valid = false;

string invalidChars = "\\/:;\*?! .,\"<>|";

while (!valid) {

cout << "Введіть назву файлу: ";

cin >> filename;

if (filename.length() > 20) {

cout << "Невірна назва файлу. Назва файлу повинна бути не більше 20 символів." << endl;

}

else {

bool containsInvalidChars = false;

for (char c : filename) {

if (invalidChars.find(c) != string::npos) {

containsInvalidChars = true;

break;

}

}

if (containsInvalidChars) {

cout << "Назва файлу не може містити наступні символи " << invalidChars << endl;

}

else {

valid = true;

}

}

}

filename += ".txt";

return filename;

}

char\* get\_filename1() {

const int max\_filename\_length = 20;

bool valid = false;

char\* filename = new char[max\_filename\_length];

char txt[] = ".txt";

while (!valid) {

cout << "Введіть назву файлу: ";

cin >> filename;

if (strlen(filename) > max\_filename\_length - 1) {

cerr << "Невірна назва файлу. Назва файлу повинна бути не більше 20 символів." << endl;

}

else {

bool containsInvalidChars = false;

for (int i = 0; i < strlen(filename); i++) {

if (filename[i] == '\n' || filename[i] == '!' || filename[i] == ';' || filename[i] == ':' || filename[i] == ' ' || filename[i] == ',' || filename[i] == '.' || filename[i] == '/' || filename[i] == '\\' || filename[i] == ':' || filename[i] == '\*' || filename[i] == '?' || filename[i] == '"' || filename[i] == '<' || filename[i] == '>' || filename[i] == '|') {

containsInvalidChars = true;

break;

}

}

if (containsInvalidChars) {

cout << "Назва файлу не може містити наступні символи \\/:;\*?! .,\"<>|" << endl;

}

else {

valid = true;

}

}

}

char\* finalfilename = new char[strlen(filename) + strlen(txt) + 1];

strcpy(finalfilename, filename);

strcat(finalfilename, txt);

cin.clear();

return finalfilename;

}

//функція для створення нового файлу

void create\_file(const string& filename) {

ofstream file(filename);

cout << "Створено новий файл " << filename << endl;

cout << endl;

if (!file.is\_open()) {

cout << "Не вдалося створити файл " << filename << endl;

}

file.close();

}

void create\_file(char filename[]) {

FILE\* fp;

cout << "Створено новий файл " << filename << endl;

cout << endl;

fp = fopen(filename , "w");

if (fp == nullptr) {

cout << "Не вдалося створити файл " << filename << endl;

return;

}

fclose(fp);

}

//функція для запису тексту в файл

void text\_to\_file(string& filename) {

ofstream file(filename);

string line;

int max\_size = 10 \* 1024 \* 1024;

int curr\_size = 0;

int line\_count = 0;

if (!file.is\_open()) {

cout << "Не вдалося відкрити файл " << filename << endl;

return;

}

cout << "-------------------------------------------------------"<< endl;

cout << "Введіть текст, який бажаєте записати у файл для подальшого шифрування " << filename << endl;

cout << "Натисніть 1, щоб завершити запис."<< endl;

cout << endl;

while (getline(cin, line)) {

if (line == "1") {

break;

}

curr\_size += line.length();

line\_count++;

if (curr\_size > max\_size) {

cout << "Помилка вводу. Розмір файлу не має перевищувати 10 МВ" << endl;

file.close();

remove(filename.c\_str());

exit(1);

}

file << line << endl;

}

file.close();

cout << endl;

cout << "Запис завершено." << endl;

cout << "-------------------------------------------------------" << endl;

cout << endl;

cout << endl;

cout << "-------------------------------------------------------" << endl;

cout << "Вміст створеного файлу:" << endl;

}

void text\_to\_file(char\* filename) {

FILE\* fp;

fp = fopen(filename, "w");

char line[7000];

if (fp == nullptr) {

cout << "Не вдалося відкрити файл " << filename << endl;

return;

}

cout << "-------------------------------------------------------" << endl;

cout << "Введіть текст, який бажаєте записати у файл для подальшого шифрування " << filename << endl;

cout << "Натисніть 1, щоб завершити запис." << endl;

cout << endl;

while (scanf(" %[^\n]%\*c", line)) {

if (line[0] == '1' && strlen(line) == 1) {

break;

}

else if (strlen(line) > 7000) {

cout << "Рядок має бути менше 4192 символів. Спробуйте ще раз." << endl;

continue;

}

fprintf(fp, "%s \n", line);

}

fclose(fp);

cout << endl;

cout << "Запис завершено." << endl;

cout << "-------------------------------------------------------" << endl;

cout << endl;

cout << endl;

cout << "-------------------------------------------------------" << endl;

cout << "Вміст створеного файлу:" << endl;

cout << endl;

}

//функція для відкриття файлу та відображення його змісту

void open\_file(string& filename) {

ifstream file(filename);

string line;

if (file.is\_open()) {

/\* cout << endl;

cout << "-------------------------------------------------------" << endl;

cout << "Вміст файлу:" << endl;

cout << endl;\*/

while (getline(file, line)) {

cout << line << endl;

}

file.close();

cout << endl;

cout << "-------------------------------------------------------" << endl;

}

else {

cout << "Неможливо відкрити файл" << endl;

}

}

void open\_file(char\* filename) {

bool fileExists = true;

FILE\* fp = fopen(filename, "r");

char\* line = new char[4192];

/\*cout << endl;

cout << "-------------------------------------------------------" << endl;

cout << "Вміст файлу:" << endl;

cout << endl;\*/

while (fgets(line, 7000, fp)) {

printf( "%s", line);

}

fclose(fp);

delete[] line;

cout << endl;

cout << "-------------------------------------------------------" << endl;

}

//функція для введення К

int enter\_k() {

int k;

bool valid = false;

while (!valid) {

cout << "Введіть К (кількість елементів для розбиття на групи)" << endl;

cin >> k;

if (k > 20) {

cout << "К має бути не більше 20" << endl;

}

else {

valid = true;

}

}

return k;

}

//функція для зворотного запису символів в рядку

string reverse\_string(string line) {

reverse(line.begin(), line.end());

return line;

}

char\* reverse\_string(char\* line, int k) {

char\* newline = new char[k];

for (int i = 0, j = k - 1; j >= 0; i++, j--) {

newline[i] = line[j];

}

return newline;

}

//функція для шифрування тексту в рядках

string encrypt\_string(string line, int k) {

string result = "";

int n = line.length();

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

string group = line.substr(i, k);

result += reverse\_string(group);

}

return result;

}

char\* encrypt\_string(char\* line, int k, int n) {

char\* result = new char[n];

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

char\* group = new char[k];

for (int j = i; j < i + k && j < n; j++)

{

group[j - i] = line[j];

}

char\* reverse = reverse\_string(group, min(k,n - i));

for (int j = i; j < i + k && j < n; j++)

{

result[j] = reverse[j - i];

}

}

return result;

}

//функція для запису в файл шифрування тексту в рядках

void encryption(string& filename, int k) {

ifstream infile(filename);

ofstream outfile("encrypted.txt");

string line;

while (getline(infile, line)) {

string encrypted\_line = encrypt\_string(line, k);

outfile << encrypted\_line << endl;

}

infile.close();

outfile.close();

cout << endl;

//cout << "Вміст зашифрованого файлу:" << endl;

//ifstream inputfile("encrypted.txt");

//cout << inputfile.rdbuf();

//inputfile.close();

}

void encryption(char\* filename, int k) {

FILE\* inf;

FILE\* outf;

inf = fopen(filename, "r");

char encryptedfilename[] = "encrypted.txt";

outf = fopen("encrypted.txt", "w");

char line[4192];

while (fgets(line, 4192, inf)) {

int n = strlen(line) - 2;

char\* newline = encrypt\_string(line, k, n);

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

fprintf(outf, "%c", newline[i]);

}

fprintf(outf, "\n");

}

fclose(inf);

fclose(outf);

cout << endl;

//cout << "Вміст зашифрованого файлу:" << endl;

//open\_file(encryptedfilename);

}

//функція для відбору непарних рядків

void choose\_line() {

ifstream inFile("encrypted.txt");

ofstream outFile("choose\_line.txt");

if (inFile.is\_open()) {

string line;

while (getline(inFile, line)) {

if (line.length() %2 !=0) {

outFile << line << endl;

}

}

inFile.close();

outFile.close();

}

}

void choose\_line1() {

FILE\* inf;

FILE\* outf;

char inputfilename[] = "encrypted.txt";

inf = fopen(inputfilename, "r");

char outputfilename[] = "choose\_line.txt";

outf = fopen(outputfilename, "w");

char line[7000];

while (fgets(line, 7000, inf)) {

if((strlen(line) - 1 )% 2 == 1)

fprintf(outf, "%s", line);

}

fclose(inf);

fclose(outf);

}

//функція для перевірки латинських символів

bool is\_word\_latin(string word) {

for (int i = 0; i < word.length(); i++) {

if (!((word[i] >= 'a' && word[i] <= 'z' ) || (word[i] >= 'A' && word[i] <= 'Z'))) {

return false;

}

}

return true;

}

bool is\_word\_latin(char\* word, int n) {

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

if (!((word[i] >= 'a' && word[i] <= 'z') || (word[i] >= 'A' && word[i] <= 'Z'))) {

return false;

}

}

return true;

}

//функція для видалення латинських слів

void remove\_latin\_words() {

ifstream input\_file("choose\_line.txt");

ofstream output\_file("clean\_latin.txt");

string line;

while (getline(input\_file, line))

{

string final\_line = "";

string word = "";

for (int i = 0; i < line.length(); i++) {

if (line[i] == ' ') {

if (!is\_word\_latin(word)) {

final\_line += word;

final\_line += " ";

}

word = "";

}

else {

word += string(1, line[i]);

}

}

if (!is\_word\_latin(word)) {

final\_line += word;

}

output\_file << final\_line << endl;

}

input\_file.close();

output\_file.close();

cout << "-------------------------------------------------------" << endl;

cout << "Вміст зашифрованого нового файлу:" << endl;

cout << endl;

}

void remove\_latin\_words1() {

FILE\* inf;

FILE\* outf;

char inputfilename[] = "choose\_line.txt";

inf = fopen(inputfilename, "r");

char outputfilename[] = "clean\_latin.txt";

outf = fopen(outputfilename, "w");

char line[7000];

while (fgets(line, 7000, inf)) {

char\* final\_line = new char[1];

strcpy(final\_line, "");

char\* word = new char[1];

strcpy(word, "");

int n = 0;

int m = strlen(line) - 1;

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

if (line[i] == ' ') {

if (!is\_word\_latin(word,n)) {

strcat(final\_line, word);

strcat(final\_line, " ");

}

n = 0;

strcpy(word,"");

}

else {

n++;

strncat(word, &line[i], 1);

}

}

if (!is\_word\_latin(word, n)) {

strcat(final\_line, word);

}

fprintf(outf, "%s \n", final\_line);

}

fclose(inf);

fclose(outf);

cout << "-------------------------------------------------------" << endl;

cout << "Вміст зашифрованого нового файлу:" << endl;

cout << endl;

}

**Main.cpp**

#include <iostream>

#include <fstream>

#include <string>

#include <Windows.h>

#include "Files.h"

using namespace std;

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

setlocale(LC\_CTYPE, "ukr");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

if (argc != 3) {

cout << "Введіть параметр FileStream/FilePointer" << endl;

return 1;

}

char\* mode = argv[2];

if (!strcmp(mode, "FileStream")) {

cout << "Ви в режимі роботи з потоками" << endl;

cout << endl;

string filename = get\_filename();

create\_file(filename);

text\_to\_file(filename);

open\_file(filename);

int k = enter\_k();

encryption(filename, k);

choose\_line();

remove\_latin\_words();

string name = "clean\_latin.txt";

open\_file(name);

}

else if (strcmp(mode, "FilePointer") == 0) {

cout << "Ви в режимі роботи з вказівниками" << endl;

cout << endl;

int k = 0;

char\* filename = new char[1];

filename = get\_filename1();

create\_file(filename);

text\_to\_file(filename);

open\_file(filename);

k = enter\_k();

encryption(filename, k);

choose\_line1();

remove\_latin\_words1();

char name[] = "clean\_latin.txt";

open\_file(name);

}

else {

cout << "Невідомий параметр! Ви маєте ввести FileStream/FilePointer" << endl;

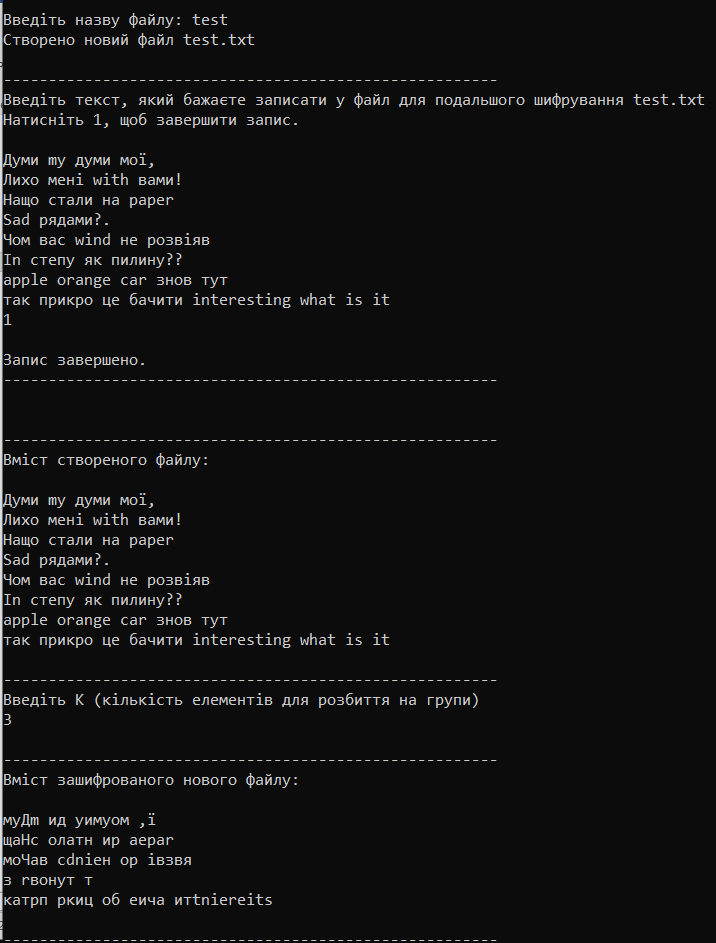
return 1;

}

return 0;

}

1. **Перевірки**

****