**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

**Факультет комп’ютерних наук**

**Кафедра інженерії програмного забезпечення**

**ЗВІТ**

*з лабораторної роботи № 3*

**«Рядки та файли»**

**Варіант № 5**

Дисципліна «Алгоритми та структури даних»

Спеціальність «Інженерія програмного забезпечення»

121–ЛР.03–208. 22120802

***Cтудент*** *\_\_\_\_\_\_\_\_\_\_****Д.М.Дзундза***

*(підпис)*

*\_\_03.10.2021 \_\_*

*(дата)*

***Викладач*** *\_****\_\_\_\_\_\_\_\_К. О. Кірей***

*(підпис)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**Миколаїв – 2021**

Лабораторна робота №3

**Тема роботи:** Бінарні файли

**Мета роботи:** ознайомитися з моделлю бінарного файлу; ознайомитися зі способами доступу в довічним файлі; навчитися застосовувати отримані знання для вирішення практичних завдань.

**Завдання:**

Завдання 1.

Реалізуйте рішення задач 3.1-3.4 з використанням мови С ++.

Задача 3.1. Друк питальних речень

Написати програму, що зчитує текст з файлу Shakespeare\_Winter'sTale.txt і виводить з цього тексту на екран тільки питальні речення. Файл Shakespeare\_Winter'sTale.txt містить текст п'єси У. Шекспіра «Зимова казка».

Організуйте вивід кожного питального речення задачі 3.1 в інший файл.

Задача 3.2. Записати дамп файлу

Записати і переглянути, як виглядає двійковий дамп файлу, записаного наступною програмою: у файл Z3.dat записуються дві змінні типу double, потім пишеться масив елементів типу int, а потім

другий масив – типу char. Далі обчислюється розмір записаного файлу і дозаписується в цей же файл.

Запишіть дамп файлу задачі 3.1.

Задача 3.3. Редагування масиву цілих чисел у бінарному файлі

Написати програму, яка зчитує в бінарний файл статичний масив цілих чисел розмірністю n. Потім користувач повинен мати можливість обрати будь-який елемент масиву і вказати його нове

значення, що має записатися у файл замість старого.

Написати програми для редагування елементів масиву для: динамічного одномірного масиву; двомірного масиву. Для одномірного масиву додати можливість не тільки редагування, а й видалення елементів масиву.

Задача 3.4. Порівняння файлів

Написати програму, що порівнює два файли за розміром або за вмістом. Програма має визначати: повну ідентичність двох файлів або відміну за розміром або вмістом. Перевірте, як працює програма для файлів різних форматів. Реалізуйте програму так, щоб ще виконувалося порівняння за том, чи є один з файлів частиною іншого.

Задача 3.5

Варіант 5

Написати програму, що порівнює два текстові файли і формує

результуючий файл з незбіжних фрагментів. Крім цього, програма має

обчислювати і виводити на екран відсоток першого та другого файлу в

результуючому файлі.

Текст програми:

Лістинг 1 – Вивід питальних речень

//3.1,3.2.2

#include "windows.h"

#include <fstream>

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main()

{

ifstream in("D://Shakespeare\_WinterTale.txt", ios::binary);//read file

ofstream out("questions.txt", ios::binary);//Create file with questions

int numOfQuestions = 0;//Number of questions

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

cout << "Couldn't open file on read!" << endl;

in.close();

return 1;

}

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

cout << "Couln't open file on write!" << endl;

out.close();

return 1;

}

//Num of chars

in.seekg(0, ios::end);

int length = in.tellg();

in.seekg(0, ios::beg);

//Declare space for text

char\* text = new char[length + 1];

in.read(text, length);

text[length] = '\0';

int startOfSentence = 0;

int i = 0;

string sentence = "";

//find end of sentence

for (i = startOfSentence; i < length; i++) {

sentence += text[i];

if (text[i] == '.' || text[i] == '?' || text[i] == '!' || (text[i] == '.' && text[i + 1] == '.' && text[i + 2] == '.')) {//Кінець речення

if (text[i] == '.' && text[i + 1] == '.' && text[i + 2] == '.') {

startOfSentence = i + 3; // if "...", skip 3

}

else {

startOfSentence = i + 1; // if ?,!,. skip 1

}

if (sentence[sentence.length() - 1] == '?') {

numOfQuestions++;

out.write(sentence.c\_str(), sentence.length());

}

sentence = "";

}

}

cout << "Num of questions: " << numOfQuestions << endl;

delete[]text;

in.close();

in.clear();

out.close();

out.clear();

//Dump file

in.open("questions.txt", ios::binary);

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

cout << "Couldn't open file for read dump!" << endl;

return 1;

}

//file length

in.seekg(0, ios::end);

length = in.tellg();

in.seekg(0, ios::beg);

//text from file

char\* textFromFile = new char[length + 1];

textFromFile[length] = '\n';

//Reading text

in.read(textFromFile, length);

in.close();

ofstream dumpFile("dump.dat", ios::binary);

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

cout << "Couldn't open file for write dump!" << endl;

return 1;

}

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

dumpFile.write((char\*)&textFromFile[j], sizeof(textFromFile[j]));

}

//dumpFile.write(textFromFile, length);

dumpFile.close();

cout << "Dump succesfully written!" << endl;

delete[]textFromFile;

}

Лістинг 2 – Запис дамбу чисельних значень та літер

//3.2.1

#include <iostream>

#include <fstream>

#include <locale.h>

#include <windows.h>

using namespace std;

int main() {

int arr1[] = { 0, 10, 4, 25, 1000, 14, 0, 47, 11, 1 };

char arr2[] = { 'a', '\ 0', 'b' };

double m = 3.14, n = 0.0003;

// відкриття для запису

ofstream fin1("Z3.dat", ios::binary);

if (!fin1.is\_open()) { // перевірка

cout << "Помилка відкриття файлу!" << endl;

return 1;

}

// Запис 8 Байт - розмір m

fin1.write((char\*)&m, sizeof(m));

// Запис 8 Байт - розмір n

fin1.write((char\*)&n, sizeof(n));

// Запис 4Байт\*10 - розмір масиву arr1

for (int i = 0; i < 10; i++) {

fin1.write((char\*)&arr1[i], sizeof(arr1[i]));

}

// Запис 1Байт \*3 - розмір масиву arr2

for (int i = 0; i < 3; i++) {

fin1.write((char\*)&arr2[i], sizeof(arr2[i]));

}

fin1.close(); // закриття файлу

// відкриття для читання

ifstream fout("Z3.dat", ios::binary);

if (!fout.is\_open()) { // перевірка

cout << "Помилка відкриття файлу!" << endl;

return 1;

}

fout.seekg(0, ios::end); // перехід в кінець файлу

// поточна позиція в байтах, виміряна від початку

long len = fout.tellg();

cout << len << endl;

fout.close(); // закриття файлу

// відкриття для дозапису в кінець файлу

ofstream fin2("Z3.dat", ios::binary | ios::app);

if (!fin2.is\_open()) {

cout << " Помилка відкриття файлу !" << endl;

return 1;

}

// Запис 4Б - розмір len

fin2.write((char\*)&len, sizeof(len));

fin2.close(); // закриття файлу

system("pause");

}

Лістинг 3 – Зміна 1D та 2D масивів

//3.3

#include "windows.h"

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <ctime>

using namespace std;

const int ROWS = 5;

const int COLS = 5;

bool isNumber(const string& str);

void printArr(int arr[], int size);

void resetToZero(int arr[], int size);

void fillArr(int arr[], int size);

void printArr(int arr[ROWS][COLS]);

void resetToZero(int arr[ROWS][COLS]);

void fillArr(int arr[ROWS][COLS]);

int main() {

srand(time(NULL));

// Input a type of array

int type = -1;

string number;

while (type < 0 || type > 2) {

cout << "Choose type of array u want to work with:" << endl;

cout << "\t1 - 1d array" << endl;

cout << "\t2 - 2d array" << endl;

cin >> number;

if (!isNumber(number)) continue;

type = atoi(number.c\_str());

}

number = "a";

//----------1D Array---------//

if (type == 1) {

ofstream out("array.txt", ios::binary);

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

cout << "Couldnt open file on write!" << endl;

}

//Array size

int sizeOfArray = -1;

while (sizeOfArray < 0) {

cout << "Enter size of array:" << endl;

cin >> number;

if (!isNumber(number)) continue;

sizeOfArray = atoi(number.c\_str());

}

number = "a";

cout << endl;

//Array we will write to file

int\* arr = new int[sizeOfArray];

fillArr(arr, sizeOfArray);

//Write random values to array

out.write((char\*)arr, sizeof(int) \* sizeOfArray);

out.close();

delete[]arr;

//Creating array

int\* control = new int[sizeOfArray];

int answer = -1;

while (answer < 0 || answer > 3) {

cout << "Enter action u want to do with elements:" << endl;

cout << "\t1 - Edit" << endl;

cout << "\t2 - Delete" << endl;

cout << "\t3 - Add" << endl;

cin >> number;

if (!isNumber(number)) continue;

answer = atoi(number.c\_str());

}

number = "a";

if (answer == 1) {

//Open 1d array file

fstream file("array.txt", ios::binary | ios::in | ios::out);

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

cout << "Couldnt open file on read and write!" << endl;

}

else {

int index = -1;// index of element

int newValue;//new value of element

int newPosition;//position for element to change in file

//array that we have for changes

file.read((char\*)control, sizeof(int) \* sizeOfArray);

cout << "Array in file:" << endl;

printArr(control, sizeOfArray);

resetToZero(control, sizeOfArray);

cout << endl;

//Edits in file

while (index < 0 || index >= sizeOfArray) {

cout << "Enter index of element (0 to " << sizeOfArray << "):" << endl;

cin >> number;

if (!isNumber(number)) continue;

index = atoi(number.c\_str());

}

number = "a";

while (!isNumber(number)) {

cout << "Enter new value of element:" << endl;

cin >> number;

if (!isNumber(number)) continue;

newValue = atoi(number.c\_str());

}

number = "a";

cout << endl;

//Moving on position

newPosition = index \* sizeof(int);

file.seekg(newPosition, ios::beg);

//New values writing

file.write((char\*)&newValue, sizeof(int));

file.seekg(0, ios::beg);

//Array that we store after changes

file.read((char\*)control, sizeof(int) \* sizeOfArray);

cout << "Array after changes:" << endl;

printArr(control, sizeOfArray);

cout << endl;

file.close();

delete[]control;

}

}

else if (answer == 2) {

//Open file on read

ifstream fileForRead("array.txt", ios::binary);

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

cout << "Couldnt open file on read!" << endl;

}

else {

int index = -1;//Enter index of element we want to delete

//Array that we store before changes

fileForRead.read((char\*)control, sizeof(int) \* sizeOfArray);

fileForRead.close();

cout << "Array in file:" << endl;

printArr(control, sizeOfArray);

cout << endl;

//Delete element

while (index < 0 || index >= sizeOfArray) {

cout << "Enter index of element (0 to " << sizeOfArray << "):" << endl;

cin >> number;

if (!isNumber(number)) continue;

index = atoi(number.c\_str());

}

number = "a";

cout << endl;

//Array that we changed

int\* arrForWrite = new int[sizeOfArray - 1];

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

if (i < index) {

arrForWrite[i] = control[i];

}

else if (i >= index) {

arrForWrite[i] = control[i + 1];

}

}

resetToZero(control, sizeOfArray);

//Writing to file

ofstream fileForWrite("array.txt", ios::binary);

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

cout << "Couldnt open file on write!" << endl;

}

else {

fileForWrite.write((char\*)arrForWrite, (sizeOfArray - 1) \* sizeof(int));

fileForWrite.close();

}

//Checking on array

ifstream ready("array.txt", ios::binary);

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

cout << "Couldnt open file on write!" << endl;

}

else {

ready.seekg(0, ios::end);

int length = ready.tellg();

ready.seekg(0, ios::beg);

int\* rez = new int[length];

ready.read((char\*)rez, length);

cout << "Array after changes:" << endl;

printArr(rez, length / sizeof(int));

cout << endl;

delete[]rez;

ready.close();

}

delete[]control;

delete[]arrForWrite;

}

}

else {

//Open file on read

ifstream fileForRead("array.txt", ios::binary);

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

cout << "Couldnt open file on read!" << endl;

}

else {

int index = -1;//Enter index of element we want to delete

int newValue;

//Array that we store before changes

fileForRead.read((char\*)control, sizeof(int) \* sizeOfArray);

fileForRead.close();

cout << "Array in file:" << endl;

printArr(control, sizeOfArray);

cout << endl;

//Add element

while (!isNumber(number)) {

cout << "Enter new value of element:" << endl;

cin >> number;

if (!isNumber(number)) continue;

newValue = atoi(number.c\_str());

}

number = "a";

cout << endl;

//Array that we changed

int\* arrForWrite = new int[sizeOfArray + 1];

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

arrForWrite[i] = control[i];

}

arrForWrite[sizeOfArray] = newValue;

resetToZero(control, sizeOfArray);

//Writing to file

fstream fileForWrite("array.txt", ios::binary | ios::in | ios::out);

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

cout << "Couldnt open file on write!" << endl;

}

else {

fileForWrite.seekg(0, ios::end);

fileForWrite.write((char\*)&newValue, sizeof(int));

fileForWrite.close();

}

//Checking on array

ifstream ready("array.txt", ios::binary);

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

cout << "Couldnt open file on write!" << endl;

}

else {

ready.seekg(0, ios::end);

int length = ready.tellg();

ready.seekg(0, ios::beg);

int\* rez = new int[length];

ready.read((char\*)rez, length);

cout << "Array after changes:" << endl;

printArr(rez, length / sizeof(int));

cout << endl;

delete[]rez;

ready.close();

}

delete[]control;

delete[]arrForWrite;

}

}

}

else {//-----------2D file---------//

ofstream out("array2d.txt", ios::binary);

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

cout << "Couldnt open file on write!" << endl;

}

else {

//Array that we will write

int arr[ROWS][COLS];

fillArr(arr);

//Array for checking results

int control[ROWS][COLS];

//char test[2][2] = { {'a','b'}, {'c', 'd'} };

//Writing array in file

out.write((char\*)arr, ROWS \* COLS \* sizeof(int));

out.close();

//Open file on read

fstream file("array2d.txt", ios::binary | ios::in | ios::out);

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

cout << "Couldn't open file on read or write!" << endl;

}

else {

//МАСИВ, ЩО ЗБЕРІГАЄТЬСЯ У ФАЙЛІ ДО ВНЕСЕННЯ ЗМІН

file.read((char\*)&control, ROWS \* COLS \* sizeof(int));

cout << "Array in file:" << endl;

printArr(control);

resetToZero(control);

cout << endl;

//Edit file

int index1 = -1;//first index

int index2 = -1;//second index

int newValue;//new value of element

int newPosition;//position of coursor

while (index1 < 0 || index1 >= ROWS) {

cout << "Enter first index of Row (0 to " << ROWS << "):" << endl;

cin >> number;

if (!isNumber(number)) continue;

index1 = atoi(number.c\_str());

}

number = "a";

while (index2 < 0 || index2 >= COLS) {

cout << "Enter second index of Column(0 to " << COLS << "):" << endl;

cin >> number;

if (!isNumber(number)) continue;

index2 = atoi(number.c\_str());

}

number = "a";

while (!isNumber(number)) {

cout << "Enter new value of element:" << endl;

cin >> number;

newValue = atoi(number.c\_str());

}

number = "a";

cout << endl;

//Moving coursor

newPosition = index1 \* COLS \* sizeof(int) + index2 \* sizeof(int);

file.seekg(newPosition, ios::beg);

//New value writing

file.write((char\*)&newValue, sizeof(int));

file.seekg(0, ios::beg);

//Array after changes

file.read((char\*)control, sizeof(int) \* ROWS \* COLS);

cout << "Array after changes:" << endl;

printArr(control);

cout << endl;

file.close();

}

}

}

}

//2D Array

void fillArr(int arr[ROWS][COLS]) {

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

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

arr[i][j] = rand() % 9 + 1;

}

}

}

void resetToZero(int arr[ROWS][COLS]) {

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

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

arr[i][j] = 0;

}

}

}

void printArr(int arr[ROWS][COLS]) {

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

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

cout << arr[i][j] << '\t';

}

cout << endl;

}

cout << endl;

}

//1D Array

void fillArr(int arr[], int size) {

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

arr[i] = rand() % 9 + 1;

}

}

void resetToZero(int arr[], int size) {

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

arr[i] = 0;

}

}

void printArr(int arr[], int size) {

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

cout << arr[i] << " ";

}

cout << endl;

}

//Check if value is digit

bool isNumber(const string& str) {

int digits = 0; //Num of digits in a number

for (auto& element : str) {

if (isdigit(element) || element == '-')

{

digits++;

}

}

if (digits == str.length()) {

return true;

}

else {

return false;

}

}

Лістинг 4 – Порівняння файлів за змістом

//3.4

#include "windows.h"

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <ctime>

using namespace std;

int main()

{

//Українська мова в консолі

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

int quantityOfFiles = 2;//кількість файлів

string path;//шлях до файлу

ifstream\* file = new ifstream[quantityOfFiles];//масив файлів для порівняння

//ЗАДАННЯ ВІДПОВІДНИХ ФАЙЛІВ

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

cout << "Введіть шлях до файлу " << i << " :" << endl;

cin >> path;

file[i].open(path, ios::binary | ios::out);

if (!file[i].is\_open()) {

cout << "Помилка відкриття файлу!" << endl;

i--;

}

}

//ПОРІВНЯННЯ ФАЙЛІВ

//Довжина першого файла

file[0].seekg(0, ios::end);

int lengthOfFirst = file[0].tellg();

file[0].seekg(0, ios::beg);

//Довжина другого файла

file[1].seekg(0, ios::end);

int lengthOfSecond = file[1].tellg();

file[1].seekg(0, ios::beg);

//Вміст цих файлів

char\* content1 = new char[lengthOfFirst];

char\* content2 = new char[lengthOfSecond];

//Зчитування вмісту

file[0].read(content1, lengthOfFirst);

file[1].read(content2, lengthOfSecond);

//Порівняння

if (lengthOfFirst != lengthOfSecond) {

int i = 0;

int j = 0;

while (i < lengthOfFirst && j < lengthOfSecond) {

if (content1[i] != content2[j]) {

cout << "Різні за розміром та вмістом файли!" << endl;

break;

}

else {

i++;

j++;

}

}

if (i == lengthOfFirst - 1 || j == lengthOfSecond - 1) {

cout << "Один файл є частиною іншого!" << endl;

}

}

else {

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

if (content1[i] != content2[i]) {

cout << "У файлів різний вміст та однаковий розмір!" << endl;

break;

}

}

cout << "Однакові за розміром та за вмістом файли!" << endl;

}

delete[]content1;

delete[]content2;

return 0;

}

Лістинг 5 – Порівняння файлів у процентному співвідношенні

//3.5

#include "windows.h"

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <ctime>

using namespace std;

int main()

{

//Українська мова в консолі

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

int quantityOfFiles = 2;//кількість файлів

string path;//шлях до файлу

ifstream\* file = new ifstream[quantityOfFiles];//масив файлів для порівняння

//ЗАДАННЯ ВІДПОВІДНИХ ФАЙЛІВ

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

cout << "Введіть шлях до файлу:" << endl;

cin >> path;

file[i].open(path, ios::binary | ios::out);

if (!file[i].is\_open()) {

cout << "Помилка відкриття файлу!" << endl;

i--;

}

}

//ПОРІВНЯННЯ ФАЙЛІВ

//Довжина першого файла

file[0].seekg(0, ios::end);

int lengthOfFirst = file[0].tellg();

file[0].seekg(0, ios::beg);

//Довжина другого файла

file[1].seekg(0, ios::end);

int lengthOfSecond = file[1].tellg();

file[1].seekg(0, ios::beg);

//Вміст цих файлів

char\* content1 = new char[lengthOfFirst];

char\* content2 = new char[lengthOfSecond];

//Зчитування вмісту

file[0].read(content1, lengthOfFirst);

file[1].read(content2, lengthOfSecond);

//Масив, що зберігає незбіжні фрагменти

vector<char> different;

//Порівняння

int partOfFirst = 0;

int partOfSecond = 0;

if (lengthOfFirst != lengthOfSecond) {

int i = 0;

int j = 0;

while (i < lengthOfFirst && j < lengthOfSecond) {

if (content1[i] != content2[j]) {

different.push\_back(content1[i]);

different.push\_back(content2[j]);

partOfFirst++;

partOfSecond++;

i++;

j++;

}

else {

i++;

j++;

}

}

if (i < lengthOfFirst - 1) {

while (i < lengthOfFirst)

{

different.push\_back(content1[i]);

partOfFirst++;

i++;

}

}

else if (j < lengthOfSecond - 1) {

while (j < lengthOfFirst)

{

different.push\_back(content2[j]);

partOfSecond++;

j++;

}

}

}

else {

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

if (content1[i] != content2[i]) {

different.push\_back(content1[i]);

different.push\_back(content2[i]);

partOfFirst++;

partOfSecond++;

}

}

}

//Запис у новий файл

ofstream result("result.txt", ios::binary);

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

cout << "Помилка відкриття файлу!" << endl;

}

else {

char\* rez = new char[partOfFirst + partOfSecond];//інакше не записувало

for (int i = 0; i < partOfFirst + partOfSecond; i++) {

rez[i] = different[i];

}

result.write(rez, partOfFirst + partOfSecond);

result.close();

}

cout << partOfFirst << endl;

cout << partOfSecond << endl;

//Виведення відсотків

int sum = partOfFirst + partOfSecond;

int persentFirst = (float)partOfFirst \* 100 / sum;

int persentSecond = (float)partOfSecond \* 100 / sum;

cout << "Відсоток першого файлу: " << persentFirst << endl;

cout << "Відсоток другого файлу: " << persentSecond << endl;

delete[]content1;

delete[]content2;

file[0].close();

file[1].close();

return 0;

}

**Результати роботи програми:**

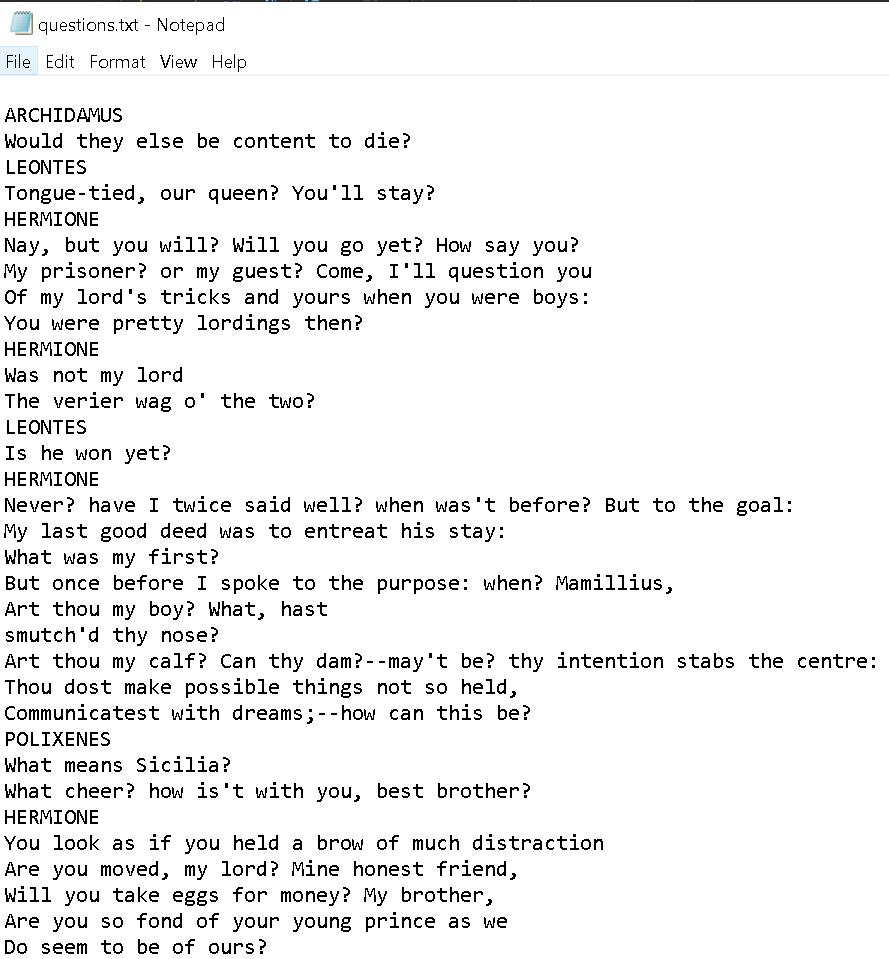


Рисунок 1 – Питальні речення

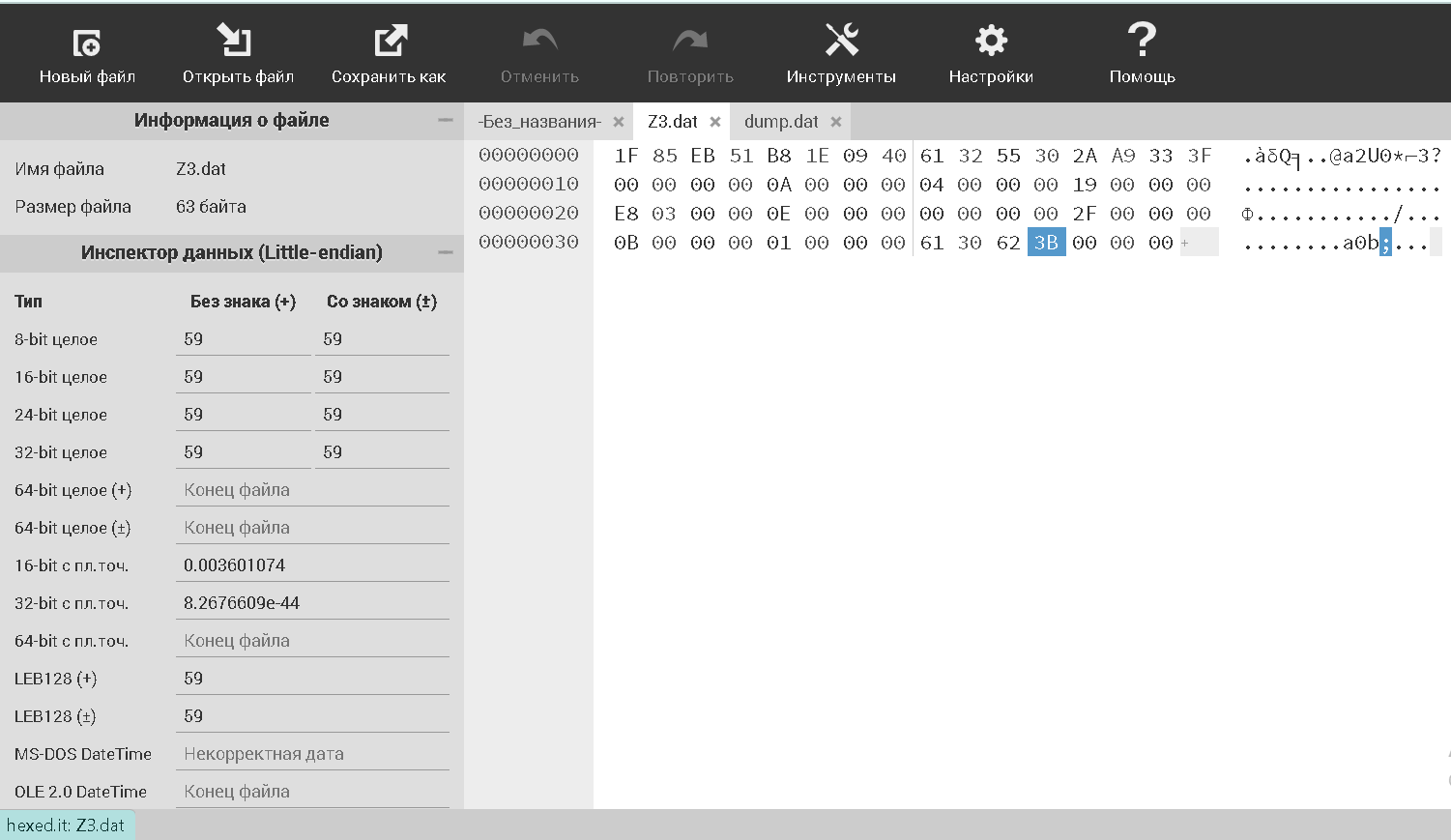


Рисунок 2 – Дамб файлу 1

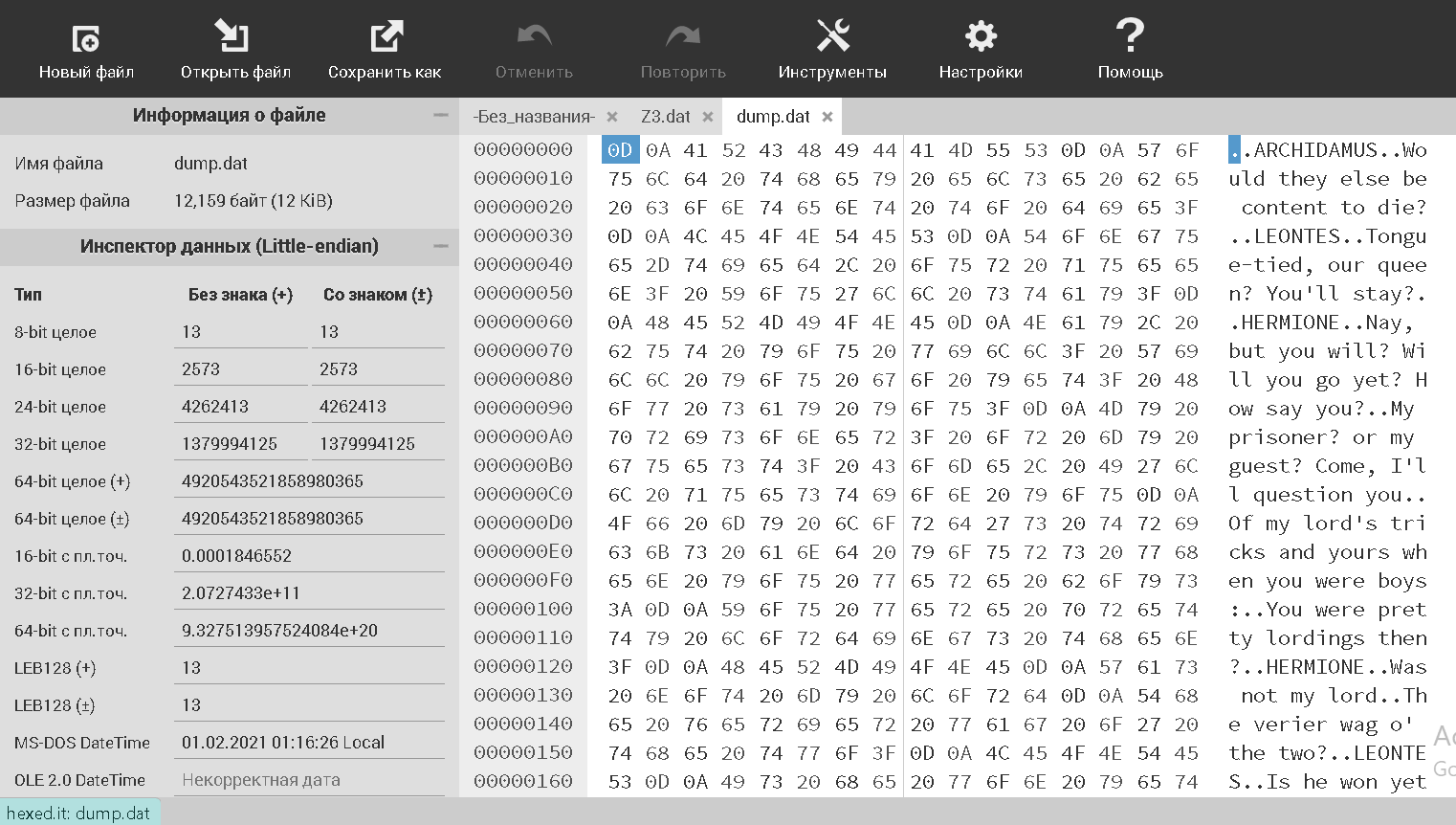


Рисунок 3 – Дамб файлу 2

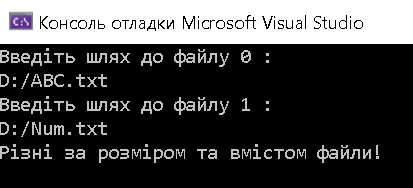


Рисунок 4 – Порівняння за змістом файлів

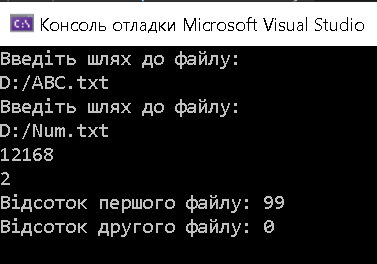


Рисунок 5 – Порівняння в процентному відношенні файлів

**Висновки:** Протягом виконання лабораторної роботи було продемонстровано способи роботи з бінарними файлами. Було продемонстровано введення-виведення та обробку даних у бінарних файлах. Було представлено вигляд бінарних файлів.