**зМІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

**Факультет комп’ютерних наук**

**Кафедра «Інженерії програмного забезпечення»**

****

**Лабораторна робота №3**

Дисципліна "Структури та організація даних"

**Виконав:**

**Студент групи 201**

Грабовський.Є.О.

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

**Викладач**

Кірей К.О.

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

**Миколаїв – 2020**

**Задача 3.1. Друк питальних речень**

**Лістинг коду:**

void firstEx() {

int length;

char\* buffer;

char separators[]{ '.','!',';' };

int separatorsLenght = 3;

std::ifstream ist("Shakespeare\_WintersTale.txt");

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

std::cout << "Opens error\n";

return;

}

length = calculateSize(ist);

buffer = new char[length];

ist.read(buffer, length);

ist.close();

for (int index = 0, start = 0, end; buffer[index] != 0; ++index) {

if (buffer[index] == '?') {

end = index + 1;

while (start != end) {

std::cout << buffer[start];

start++;

}

start++;

std::cout << '\n';

}

for (int separatorIndex = 0; separatorIndex < separatorsLenght; ++separatorIndex) {

if (separators[separatorIndex] == buffer[index]) {

start = index + 1;

}

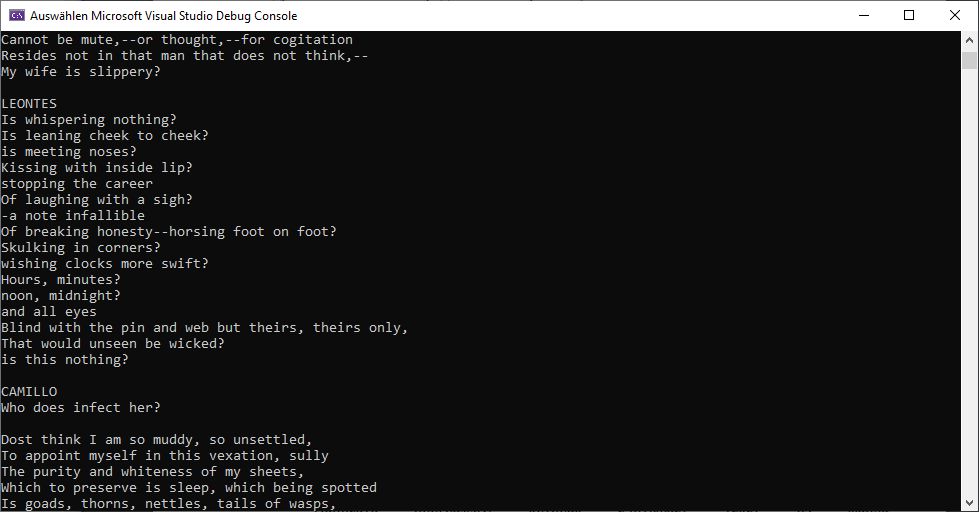
}

}

delete[] buffer;

}

**Результат виконання програми:**



**Задача 3.1. Самостійна робота**

**Лістинг коду:**

void additionFirstEx() {

int length;

char\* buffer;

char separators[]{ '.','?',';' };

int separatorsLenght = 3;

std::ifstream ist("Shakespeare\_WintersTale.txt");

std::ofstream os("text2.txt");

if (!ist.is\_open() || !os.is\_open()) {

std::cout << "Opens error\n";

return;

}

length = calculateSize(ist);

buffer = new char[length];

ist.read(buffer, length);

ist.close();

for (int index = 0, start = 0, end; buffer[index] != 0; ++index) {

if (buffer[index] == '!') {

end = index + 1;

while (start != end) {

os << buffer[start];

start++;

}

start++;

std::cout << '\n';

}

for (int separatorIndex = 0; separatorIndex < separatorsLenght; ++separatorIndex) {

if (separators[separatorIndex] == buffer[index]) {

start = index + 1;

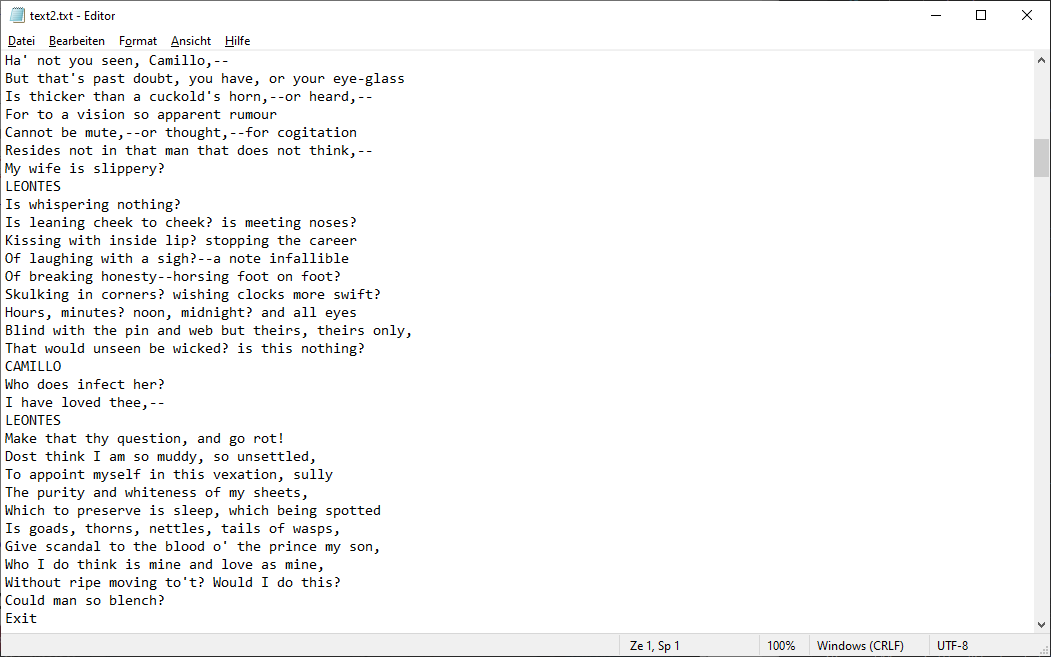
}

}

}

delete[] buffer;

}

**Результат виконання програми:**

**Задача 3.2 Запис дампу файлу**

**Лістинг коду:**

void secondEx() {

{

std::ofstream os("z3.dat", std::ios::binary);

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

std::cout << "Opens error\n";

return;

}

const int intsArrSize = 10, charsArrSize = 5;

double doubleNum1 = 36.14, doubleNum2 = 0.3543;

int intsArr[intsArrSize];

char charsArr[charsArrSize];

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

intsArr[index] = index;

for (int index = 0; index < charsArrSize; ++index) {

charsArr[index] = 70 + index;

}

os.write((char\*)&doubleNum1, sizeof(doubleNum1));

os.write((char\*)&doubleNum2, sizeof(doubleNum2));

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

os.write((char\*)&intsArr[index], sizeof(intsArr[index]));

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

os.write((char\*)&charsArr[index], sizeof(charsArr[index]));

os.close();

{

std::ifstream ist("z3.dat", std::ios::binary);

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

std::cout << "Opens error\n";

}

ist.seekg(0, std::ios::end);

int lenght = ist.tellg();

std::cout << lenght << '\n';

ist.close();

std::ofstream os("z3.dat", std::ios::binary | std::ios::app);

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

std::cout << "Opens error\n";

}

os.write((char\*)&lenght, sizeof(lenght));

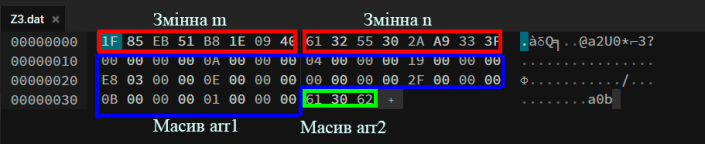
os.close();

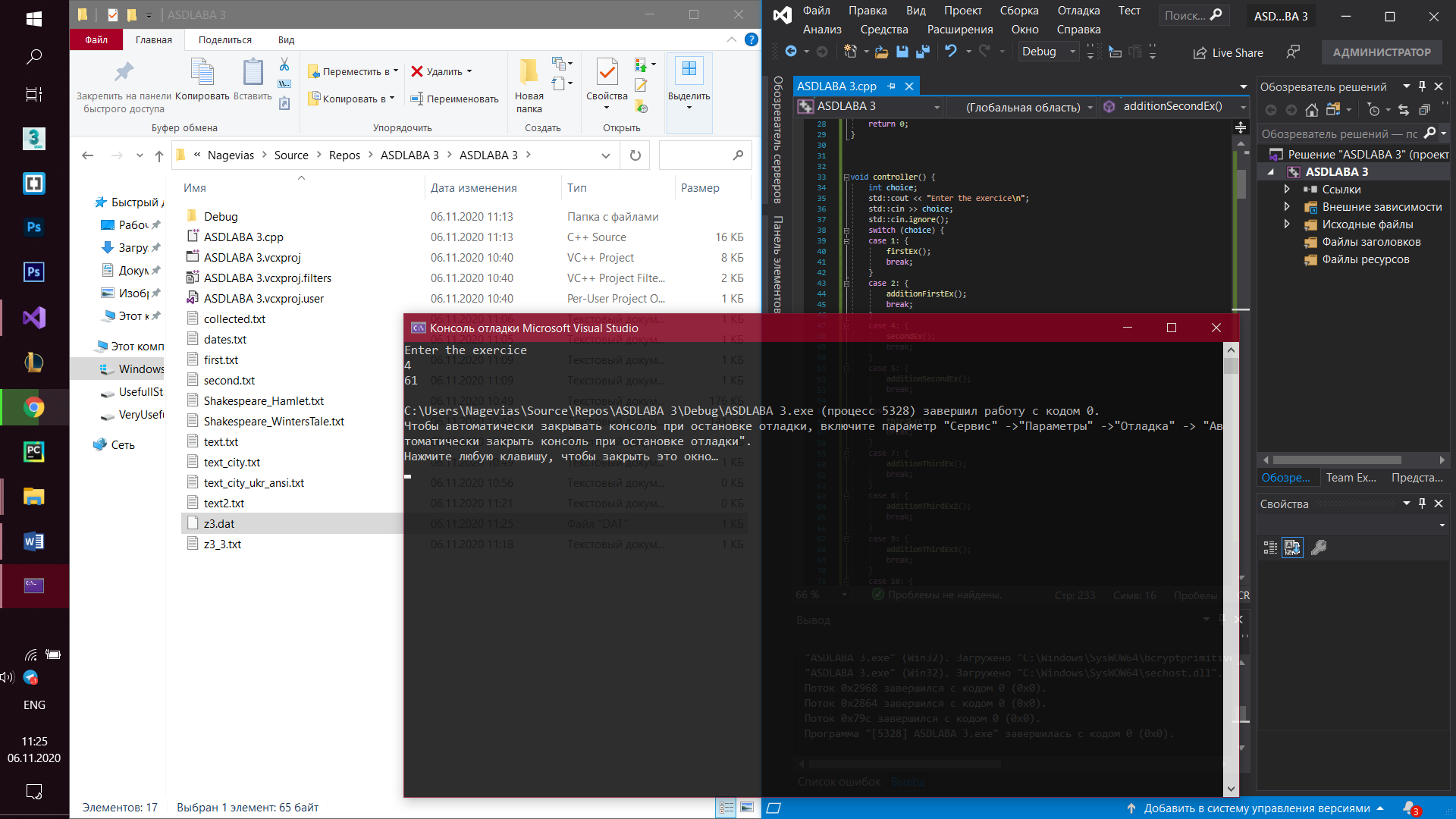
}

}

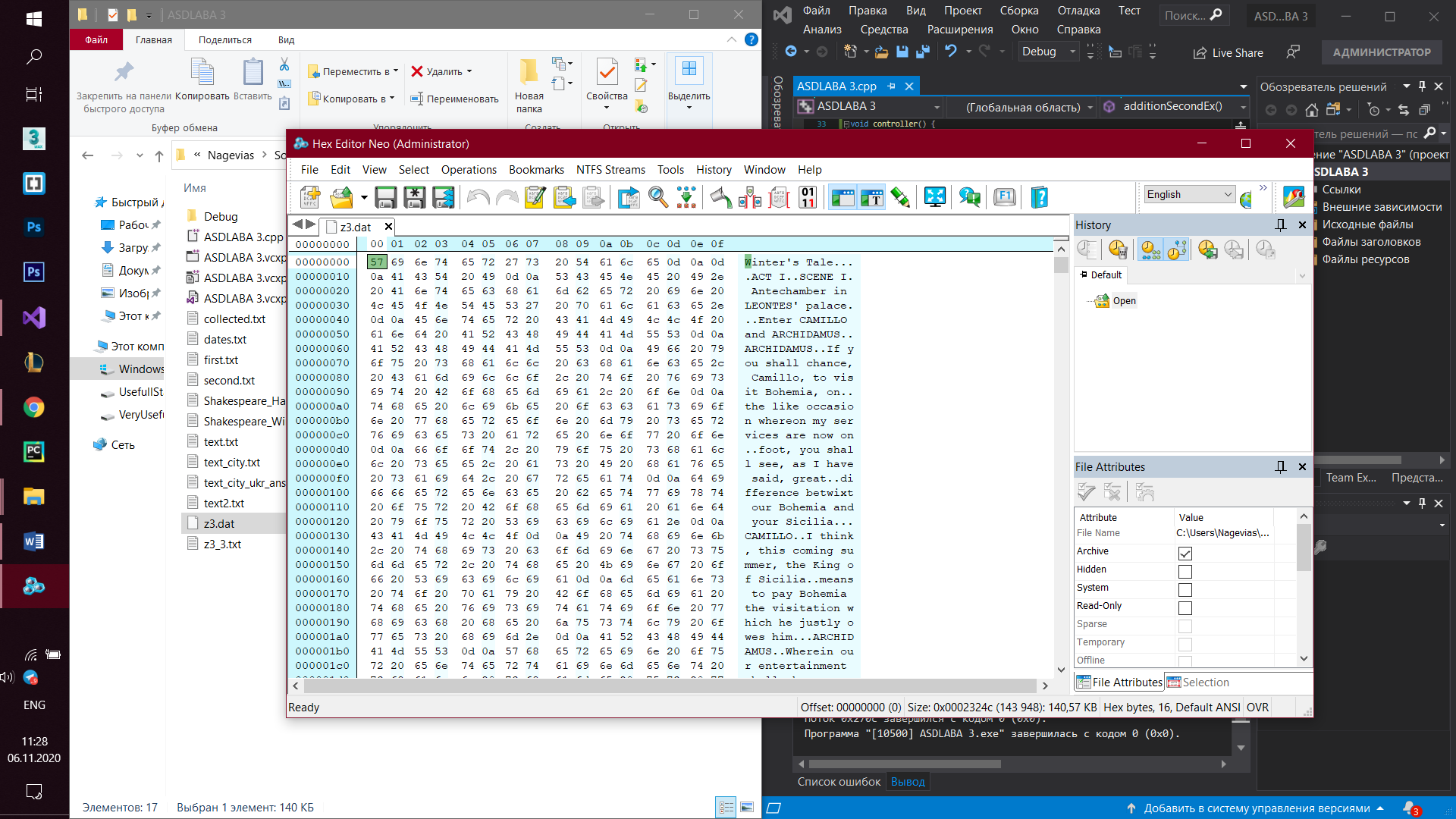
}

**Результат виконання програми:**





**Задача 3.2. Самостійна робота**



**Лістинг коду:**

void additionSecondEx() {

std::ofstream os("z3.dat", std::ios::binary);

std::ifstream ist("Shakespeare\_WintersTale.txt", std::ios::binary);

if (!(os.is\_open()) && ist.is\_open()) {

std::cout << "Opens error\n";

return;

}

int bufSize;

char\* buf;

ist.seekg(0, std::ios::end);

bufSize = ist.tellg();

ist.seekg(0, std::ios::beg);

buf = new char[bufSize];

ist.read(buf, bufSize);

os.write((char\*)buf, sizeof(char) \* bufSize);

os.close();

}

**Задача 3.3. Редагування масиву цілих чисел у бінарному файлі**

**Лістинг коду:**

void thirdEx() {

const int arrSize = 10;

int arr[arrSize];

for (int index = 0; index < arrSize; ++index) {

arr[index] = index;

}

std::ofstream ofs("z3\_3.txt", std::ios::binary);

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

std::cout << "An error has occurred\n";

return;

}

ofs.write((char\*)arr, sizeof(arr));

ofs.close();

int newNum;

int numIndex;

std::cout << "Enter the number and its index\n";

std::cin >> newNum >> numIndex;

std::fstream fst("z3\_3.txt", std::ios::binary | std::ios::in | std::ios::out);

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

std::cout << "An error has occurred\n";

return;

}

fst.seekg(0, std::ios::end);

int lenght = fst.tellg();

std::cout << "The file size ist: " << lenght << "bytes\n";

int possition = numIndex \* sizeof(int);

std::cout << "Position of the edited element: " << possition << '\n';

fst.seekp(possition, std::ios::beg);

fst.write((char\*)&newNum, sizeof(int));

fst.seekg(0);

fst.read((char\*)&arr, sizeof(arr));

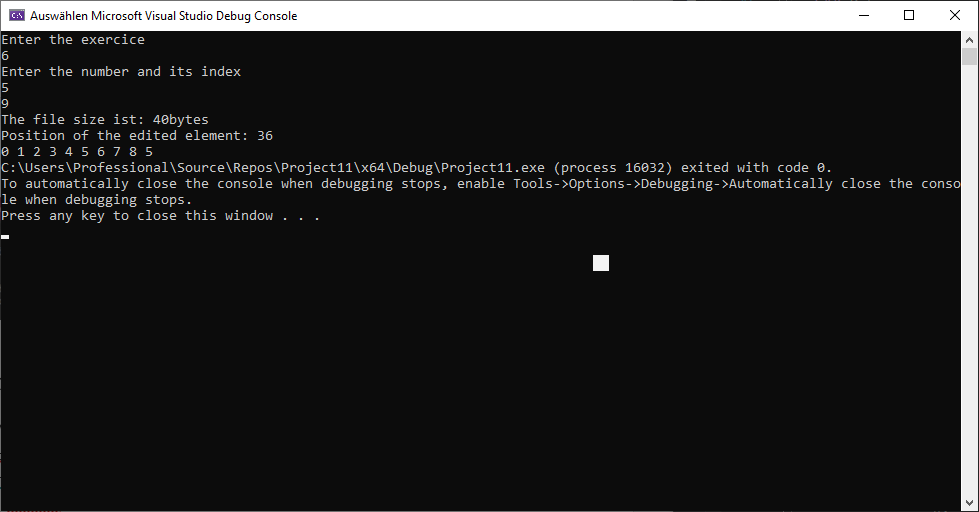
for (int index = 0; index < arrSize; ++index)

{

std::cout << arr[index] << " ";

}

**Результат виконання програми:**



**Задача 3.3. Самостійна робота (1)**

**Лістинг коду:**

void additionThirdEx() {

int arrSize;

std::cout << "Enter the size of array\n";

std::cin >> arrSize;

int\* arr = new int[arrSize];

for (int index = 0; index < arrSize; ++index) {

arr[index] = index;

}

std::ofstream ofs("z3\_3.txt", std::ios::binary);

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

std::cout << "An error has occurred\n";

return;

}

ofs.write((char\*)arr, sizeof(arr) \* arrSize);

ofs.close();

int newNum;

int numIndex;

std::fstream fst("z3\_3.txt", std::ios::binary | std::ios::in | std::ios::out);

std::cout << "Enter the number and its index\n";

std::cin >> newNum >> numIndex;

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

std::cout << "An error has occurred\n";

return;

}

fst.seekg(0, std::ios::end);

int lenght = fst.tellg();

std::cout << "The file size ist: " << lenght << "bytes\n";

int possition = numIndex \* sizeof(int);

std::cout << "Position of the edited element: " << possition << '\n';

fst.seekp(possition, std::ios::beg);

fst.write((char\*)&newNum, sizeof(int));

fst.seekp(0);

fst.read((char\*)arr, sizeof(arr) \* arrSize);

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

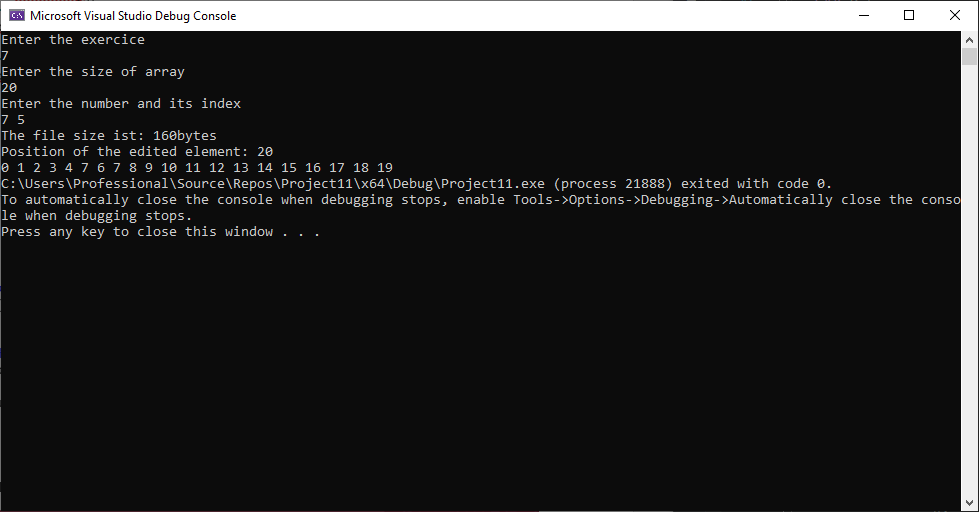
{

std::cout << arr[index] << " ";

}

}

**Результат виконання програми:**



**Задача 3.3. Самостійна робота (2)**

**Лістинг коду:**

void additionThirdEx2() {

int arrSize;

std::cout << "Enter the size of array\n";

std::cin >> arrSize;

int\*\* arr = new int\* [arrSize];

for (int index = 0; index < arrSize; ++index) {

arr[index] = new int[arrSize];

}

for (int index = 0; index < arrSize; ++index) {

for (int index2 = 0; index2 < arrSize; ++index2) {

arr[index][index2] = index2 + index \* arrSize;

std::cout << arr[index][index2] << " ";

}

std::cout << "\n";

}

std::ofstream ofs("z3\_3.txt", std::ios::binary);

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

std::cout << "An error has occurred\n";

return;

}

//ofs.write((char\*)arr, sizeof(arr));

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

ofs.write((char\*)arr[index], sizeof(arr) \* arrSize);

ofs.close();

int newNum;

int numIndex;

int numIndex2;

std::cout << "Enter the number and its index1 && index2\n";

std::cin >> newNum >> numIndex >> numIndex2;

std::fstream fst("z3\_3.txt", std::ios::binary | std::ios::in | std::ios::out);

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

std::cout << "An error has occurred\n";

return;

}

fst.seekg(0, std::ios::end);

int lenght = fst.tellg();

std::cout << "The file size ist: " << lenght << "bytes\n";

int possition = (numIndex \* arrSize \* sizeof(int)) + numIndex2 \* sizeof(int);

std::cout << "Position of the edited element: " << possition << '\n';

fst.seekp(possition, std::ios::beg);

fst.write((char\*)&newNum, sizeof(int));

fst.seekg(0);

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

fst.read((char\*)arr[index], sizeof(arr) \* arrSize);

for (int index = 0; index < arrSize; ++index) {

for (int index2 = 0; index2 < arrSize; ++index2) {

std::cout << arr[index][index2] << " ";

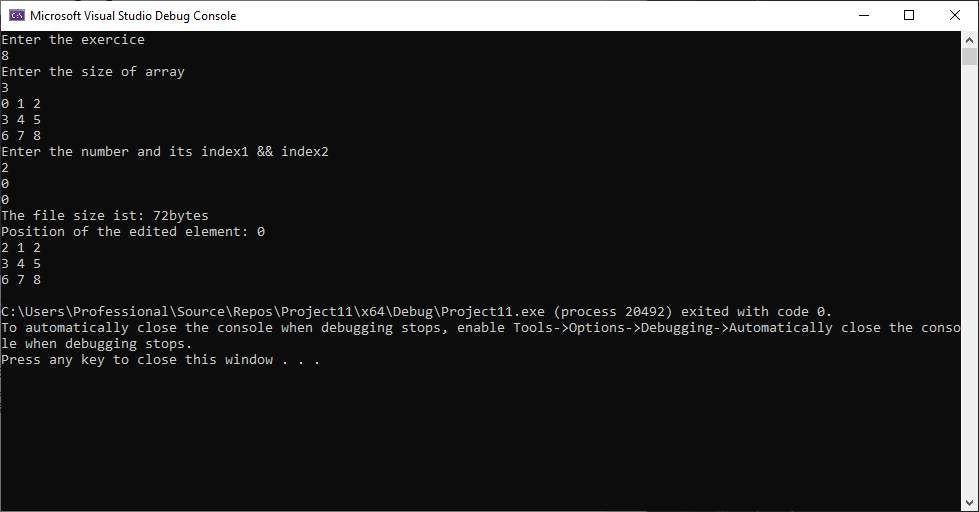
}

std::cout << "\n";

}

}

**Результат виконання програми:**



**Задача 3.3. Самостійна робота (3)**

**Лістинг коду:**

void additionThirdEx3() {

const int arrSize = 10;

int arr[arrSize];

for (int index = 0; index < arrSize; ++index) {

arr[index] = index;

}

for (int index = 0; index < arrSize; ++index) {

std::cout << arr[index] << " ";

}

std::ofstream ofs("z3\_3.txt", std::ios::binary);

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

std::cout << "An error has occurred\n";

return;

}

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

ofs.write((char\*)&arr[index], sizeof(int));

ofs.close();

int newNum;

int numIndex;

int numIndex2;

std::cout << "Enter the index you want to delete\n";

std::cin >> numIndex;

std::fstream fst("z3\_3.txt", std::ios::binary | std::ios::in | std::ios::out);

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

std::cout << "An error has occurred\n";

return;

}

fst.seekg(0, std::ios::end);

int lenght = fst.tellg();

std::cout << "The file size ist: " << lenght << "bytes\n";

int possition = (numIndex \* sizeof(int));

int tmp[arrSize - 1];

fst.seekp(possition, std::ios::beg);

for (int index = numIndex + 1; index < arrSize; ++index) {

fst.write((char\*)&arr[index], sizeof(int));

}

fst.seekg(0);

fst.read((char\*)arr, sizeof(arr));

for (int index = 0; index < arrSize - 1; ++index) {

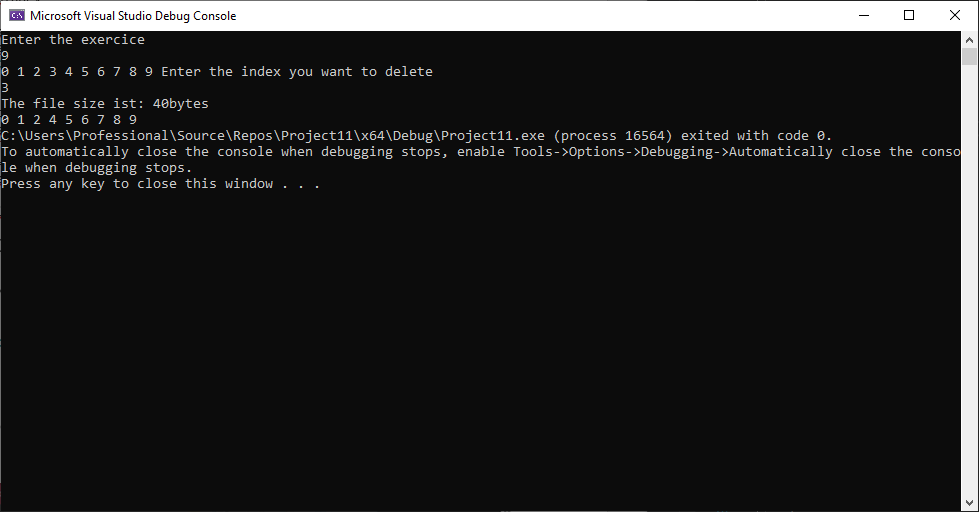
std::cout << arr[index] << " ";

}

fst.close();

}

**Результат виконання програми:**



**Задача 3.4. Порівняння файлів**

**Лістинг коду:**

void fourEx() {

std::ifstream firstFile;

std::ifstream secondFile;

firstFile.open("first.txt");

secondFile.open("second.txt");

if (!(firstFile.is\_open() && secondFile.is\_open())) {

std::cout << "The error was occure\n";

return;

}

char\* firstBuf;

char\* secondBuf;

int firstBufSize = 0;

int secondBufSize = 0;

firstBufSize = calculateSize(firstFile);

secondBufSize = calculateSize(secondFile);

if (firstBufSize != secondBufSize) {

std::cout << "Files have different sizes\n";

return;

}

firstBuf = new char[firstBufSize];

secondBuf = new char[secondBufSize];

firstFile.read((char\*)firstBuf, sizeof(int) \* firstBufSize);

secondFile.read((char\*)secondBuf, sizeof(int) \* secondBufSize);

for (int index = 0; index < firstBufSize && index < secondBufSize; ++index)

{

if (firstBuf[index] != secondBuf[index]) {

std::cout << "Files have different content\n";

return;

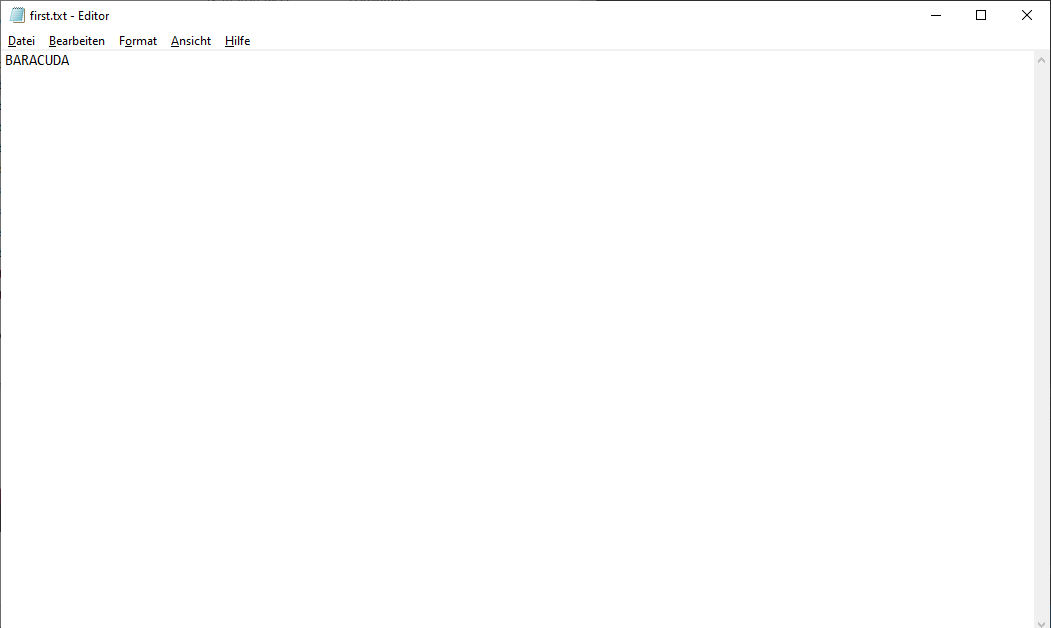
}

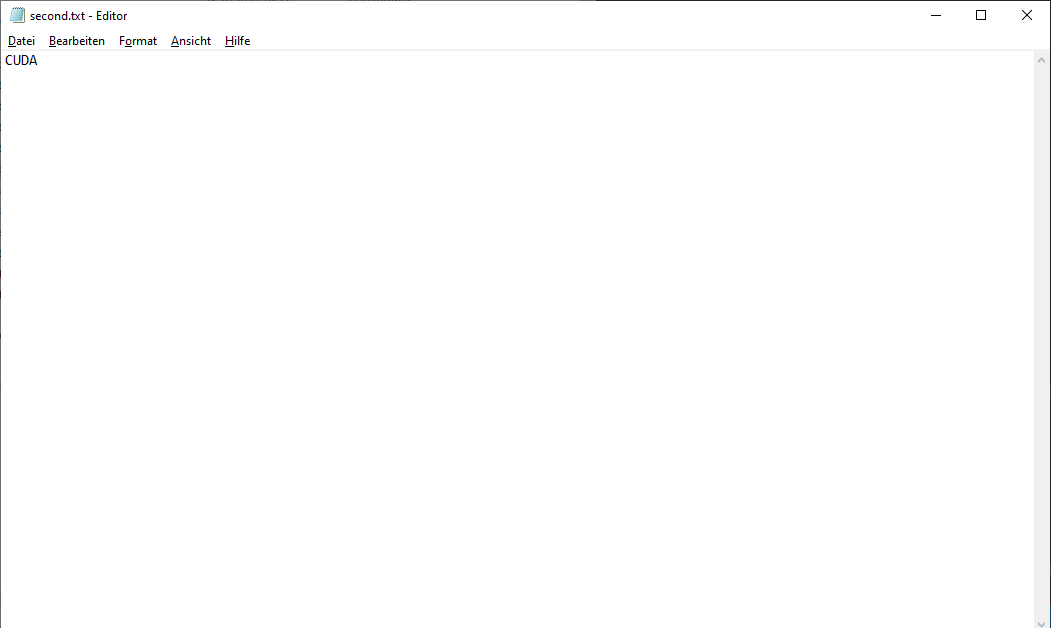
}

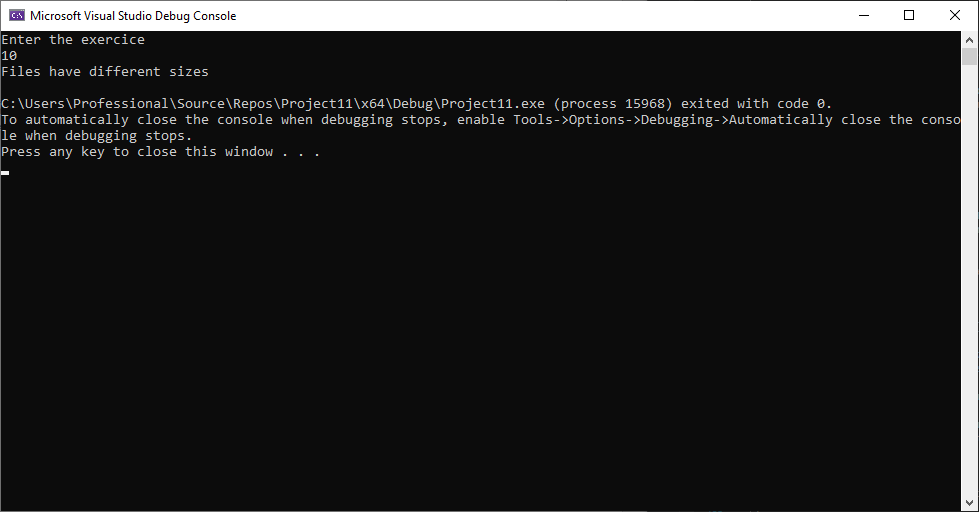
std::cout << "Files are absolutely identical\n";

}

**Результат виконання програми:**







**Задача 3.4. Самостійна робота**

**Лістинг коду:**

void additionFourEx() {

std::ifstream firstFile;

std::ifstream secondFile;

firstFile.open("first.txt");

secondFile.open("second.txt");

if (!(firstFile.is\_open() && secondFile.is\_open())) {

std::cout << "The error was occure\n";

return;

}

char\* firstBuf;

char\* secondBuf;

int firstBufSize = 0;

int secondBufSize = 0;

bool isEquals = true;

firstBufSize = calculateSize(firstFile);

secondBufSize = calculateSize(secondFile);

if (firstBufSize != secondBufSize) {

std::cout << "Files have different sizes and content\n";

isEquals = false;

}

firstBuf = new char[firstBufSize];

secondBuf = new char[secondBufSize];

firstFile.read((char\*)firstBuf, sizeof(int) \* firstBufSize);

secondFile.read((char\*)secondBuf, sizeof(int) \* secondBufSize);

if (isEquals) {

for (int index = 0; index < firstBufSize && index < secondBufSize; ++index)

{

if (firstBuf[index] != secondBuf[index]) {

std::cout << "Files have different content\n";

isEquals = false;

}

}

if (isEquals)

std::cout << "Files are absolutely identical\n";

}

bool isContainFirst, isContainSecond;

for (int index = 0, index2, index3; index < firstBufSize && index < secondBufSize; ++index) {

isContainFirst = true, isContainSecond = true;

for (index2 = 0, index3 = index; index2 < firstBufSize && index2 < secondBufSize; ++index2, ++index3) {

if (isContainFirst && firstBuf[index3] != secondBuf[index2]) {

isContainFirst = false;

if (!isContainFirst && !isContainSecond) {

break;

}

}

if (isContainSecond && secondBuf[index3] != firstBuf[index2]) {

isContainSecond = false;

if (!isContainFirst && !isContainSecond) {

break;

}

}

}

if (isContainFirst || isContainSecond) {

std::cout << "One file is part of another one\n";

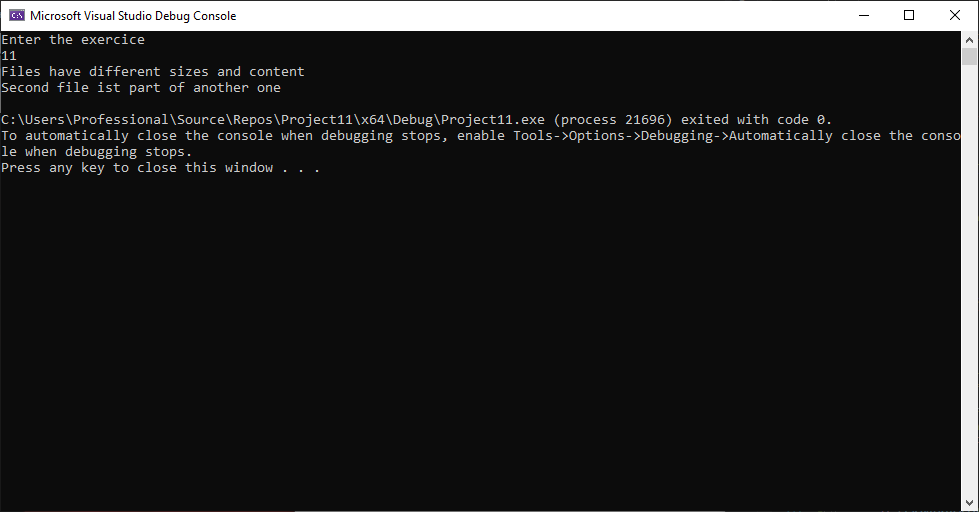
break;

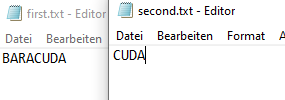
}

}

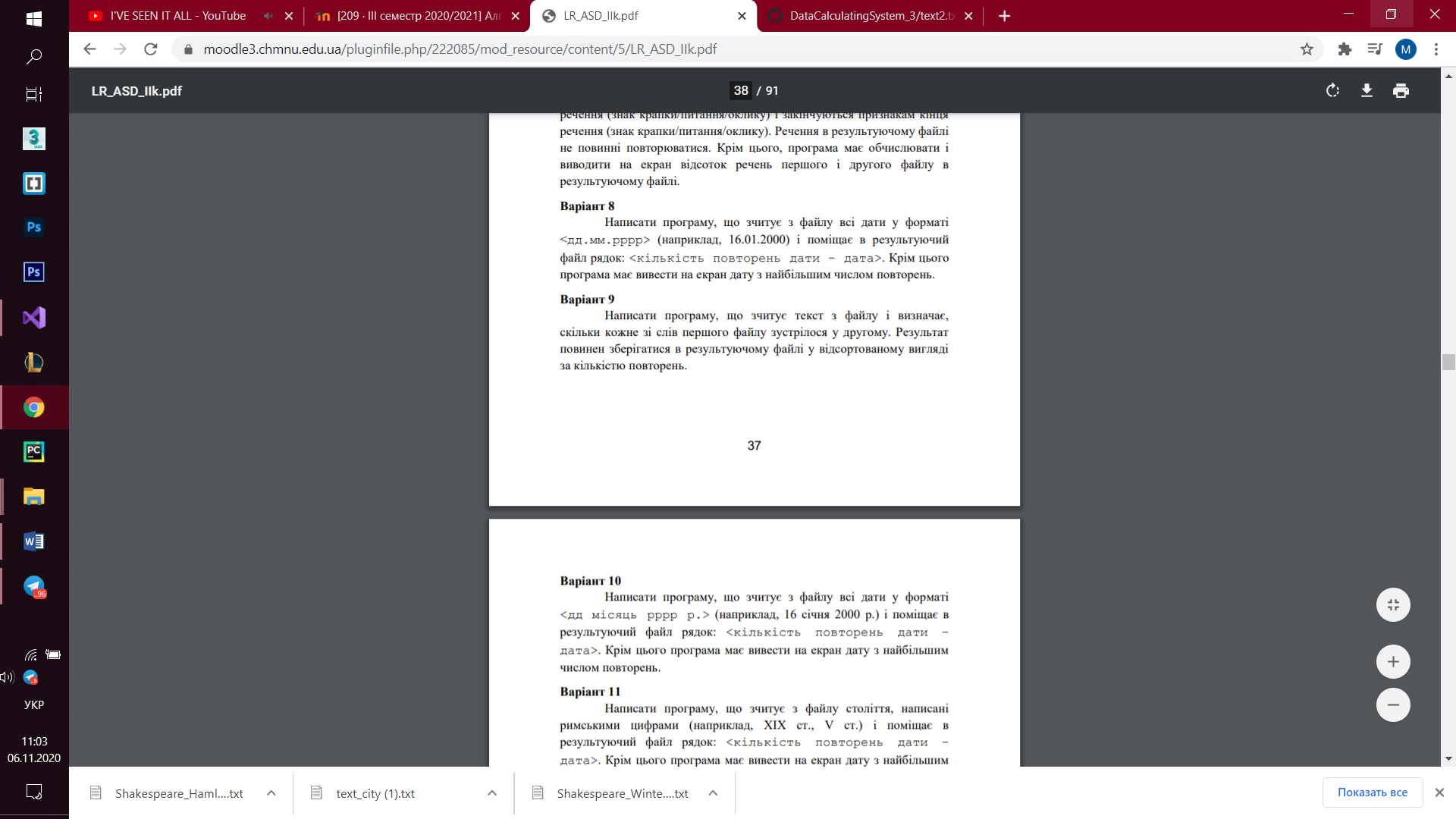
}

**Результат виконання програми:**



****

**Задача 3.5. Індивідуальне завдання**



**Лістинг коду:**

void fifth() {

std::ifstream ist("dates.txt", std::ios::binary);

std::ofstream ost("collected.txt");

if (!(ist.is\_open() && ost.is\_open())) {

std::cout << "The error has occure\n";

return;

}

std::tr1::regex rx("\\d{2}.{1}\\d{2}.{1}\\d{4}");

char\* buf;

int bufSize;

std::map<std::string, int> dates;

std::string str = "";

bufSize = calculateSize(ist);

buf = new char[bufSize];

ist.read(buf, bufSize);

buf[bufSize] = '\0';

for (int index = 0; index <= bufSize; ++index) {

if (buf[index] == ' ' || buf[index] == '\n' || buf[index] == '\r' || buf[index] == '\0') {

if (regex\_search(str.begin(), str.end(), rx)) {

dates[str]++;

}

str = "";

std::cout << std::endl;

continue;

}

std::cout << buf[index];

str += buf[index];

}

std::map<std::string, int> ::iterator it = dates.begin();

for (int index = 0; it != dates.end(); ++index, ++it) {

ost << it->first << " -> " << it->second << std::endl;

}

}

**Результат виконання програми:**

