Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

Зображення, що містить текст, Шрифт, коло, логотип

Автоматично згенерований опис

**Звіт**

про виконання

**Лабораторних та практичних робіт № 5, vns: 6, 8, 9.**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: «Файли. Системи числення. Бінарні файли. Символи та рядкові змінні. Текстові файли. Стандартна бібліотека. Створення й використання бібліотек»

***Виконав:***

студент групи ШІ-13

Орза Євгеній Сергійович

# **Тема роботи:**

Робота з створенням, читанням та редагуванням файлів.

# **Мета роботи:**

Вміння роботи з файлами через код.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Робота з файлами.
* Тема №2: Робота з бінарними файлами.
* Тема №3: Робота з стрічковими та символьними аргументами.
* Тема №4: Робота з ios флагами.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Робота з файлами.
  + Джерела Інформації
    - <https://purecodecpp.com/uk/archives/2751>
  + Що опрацьовано:
    - Створення та читання файлу.
    - Редагування файлу.
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 07.12.23
  + Звершення опрацювання теми: 09.12.23
* Тема №2: Робота з бінарними файлами.
  + Джерела Інформації:
    - <https://replace.org.ua/topic/7901/>
  + Що опрацьовано:
    - Вирішені деякі проблеми з роботою функції у бінарному файлі.
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 08.12.23
  + Звершення опрацювання теми: 08.12.2023
* Тема №3: Робота з стрічковими та символьними аргументами.
  + Джерела Інформації:
    - <https://ru.stackoverflow.com/questions/747260/%D0%A0%D0%B0%D0%B1%D0%BE%D1%82%D0%B0-%D1%81-%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%D0%BC%D0%B8-%D0%B8-%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B0%D0%BC%D0%B8-c>
  + Що опрацьовано:
    - Вирішені питання як передавати символи та стрічку у dat файл.
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 08.12.23
  + Звершення опрацювання теми: 08.12.23
* Тема №4 Робота з ios флагами.
  + Джерела Інформації:
    - <https://stackoverflow.com/questions/10359702/c-filehandling-difference-between-iosapp-and-iosate>
    - <https://cplusplus.com/reference/ios/ios/>
    - <https://learn.microsoft.com/ru-ru/cpp/standard-library/ios?view=msvc-170>
    - Що опрацьовано:
    - Опрацьовано флаги форматування для більш зручного запису у бінарні файли.
  + Статус: Ознайомлений дуже частково
  + Початок опрацювання теми: 08.12.23
  + Звершення опрацювання теми: 08.12.23

# **Виконання роботи:**

## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 VNS lab 6

* 12
* Відсортувати кожен символ за спаданням у кожному слові рядка. Створити файл з введеною стрічкою.

Завдання №2 VNS lab 8

* 12
* Створення, редагування, збереження бінарного файлу.

Завдання №3 VNS lab 9

* 12
* Копіювання файлу та підрахунок у кожній стрічці файлу голосних, вивід стрічки з найбільшою кількістю голосних.

Завдання №4 Algotester 4.1

* 2
* Сортування, видалення дублікатів та циклічний поворот масиву.
* Створено без функцій sort, uniq.

Завдання №5 Algotester 4.2

* 2
* Сортування, видалення дублікатів та циклічний поворот масиву.
* Створено з функціями sort, uniq.

Завдання №6 Algotester 6

* 2
* Створення функцій та умов для перевірки атак шахматних фігур.
* Під-час виконання завдання було порушене ментальне здоров’я виконавця та наведена порча на автора завдання.

Завдання №6 Class Practice Work

* Створення файлу, ввід тексту у файл, копіювання вмісту файлу у інший файл.

## **2. Дизайн та планована оцінка часу виконання завдань:**

Програма №1 VNS lab 6

* 1 година

Програма №2 VNS lab 8

* Рисунок 1.
* 2 години 30 хвилин

Програма №3 VNS lab 9

* 30 хвилин

Програма №4 Algotester 4.1

* 30 хвилин

Програма №5 VNS Algotester 4.2

* 30 хвилин

Програма №6 Algotester 6

* 2 години
* Нагадую за погіршення ментального здоров’я.

Програма №7 Class Practice Work

* 30 хвилин

## **3. Блок схеми:**

Завдання №2

Зображення, що містить текст, знімок екрана, схема, дизайн

Автоматично згенерований опис

Рисунок

## **4. Код програм з посиланням на зовнішні ресурси:**

Завдання №1 VNS lab 6 https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/778/files#diff-ea672d326a85342c078b86563e6cbbd3ea4b41cc573e606690bd42c5cdd95aeb

*#include* <iostream>

*#include* <algorithm>

*#include* <sstream>

*using* *namespace* std;

string sort\_string(string& *input*){

   string result;

   istringstream inputsting(*input*);

   ostringstream outputstring;

*while*(inputsting *>>* result){

    sort(result.*begin*(), result.*end*());

    outputstring *<<* result *<<* " ";

    }

*return* outputstring.*str*();

}

int main() {

    string s;

    cout *<<* "Enter a string: ";

    getline(cin, s);

    string sortedString *=* sort\_string(s);

    cout *<<* "Sorted string: " *<<* sortedString *<<* endl;

*return* *0*;

}

Код vns lab 6

Завдання №2 VNS lab 8 <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/778/files>

*#include* <iostream>

*#include* <fstream>

*#include* <vector>

*#include* <string>

*using* *namespace* std;

struct Musicdisc {

    string title;

    string author;

    int duration;

    double price;

};

void printFile(const string& *filename*){

    ifstream file (*filename*);

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

        cout *<<* "Error: unable to open file" *<<* endl;

*return*;

    }

    Musicdisc disc;

*while* (file *>>* disc.title *>>* disc.author *>>* disc.duration *>>* disc.price){

        cout *<<* "Title: " *<<* disc.title *<<* " Author:" *<<* disc.author *<<*" Duration: " *<<* disc.duration *<<* " Price:" *<<* disc.price *<<*endl;

    }

    file.*close*();

}

void addElement(const string& *filename*, const Musicdisc& *newDisc*){

    ofstream file(*filename*, ios::app);

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

        cout *<<* "Error: unable to open file" *<<* endl;

*return*;

    }

    file *<<* *newDisc*.title *<<* " " *<<* *newDisc*.author *<<* " " *<<* *newDisc*.duration *<<* " " *<<* *newDisc*.price *<<* "\n";

    file.*close*();

}

void removeElement(const string& *filename*, int *duration*) {

    ifstream file(*filename*);

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

        cout *<<* "Error: unable to open file" *<<* endl;

*return*;

    }

    ofstream tempFile("temp.dat");

*if* (*!*tempFile.*is\_open*()) {

        cout *<<* "Error: unable to open temp file for writing" *<<* endl;

        file.*close*();

*return*;

    }

    Musicdisc disc;

    bool isdeleted *=* false;

*while* (file *>>* disc.title *>>* disc.author *>>* disc.duration *>>* disc.price) {

*if* (disc.duration *==* *duration* *&&* *!*isdeleted) {

            isdeleted *=* true;

*continue*; *// Пропускаємо видалений елемент*

        }

        tempFile *<<* disc.title *<<* " " *<<* disc.author *<<* " " *<<* disc.duration *<<* " " *<<* disc.price *<<* "\n";

    }

    file.*close*();

    tempFile.*close*();

*// Видаляємо оригінальний файл та перейменовуємо тимчасовий на його місце*

    remove(*filename*.*c\_str*());

    rename("temp.dat", *filename*.*c\_str*());

}

int main (){

    const string filename *=* "music\_discs.dat";

    Musicdisc disc1 *=* { "Album1", "Artist1", *60*, *19.99* };

    Musicdisc disc2 *=* { "Album2", "Artist2", *90*, *24.99* };

    Musicdisc disc3 *=* { "Album3", "Artist3", *120*, *29.99* };

    addElement(filename, disc1);

    addElement(filename, disc2);

    addElement(filename, disc3);

    std::cout *<<* "All artist" *<<* endl;

    printFile(filename);

    removeElement(filename, *90*);

   cout *<<* "\nAfter deleting an element:" *<<* endl;

    printFile(filename);

    Musicdisc newDisc1 *=* { "NewAlbum1", "NewArtist1", *75*, *21.99* };

    Musicdisc newDisc2 *=* { "NewAlbum2", "NewArtist2", *80*, *27.99* };

    addElement(filename, newDisc1);

    addElement(filename, newDisc2);

    cout *<<* "\nAfter adding new elements:" *<<* endl;

    printFile(filename);

*return* *0*;

}

Код vns lab 8

Завдання №3 Програма №3 VNS lab 9

*#include* <iostream>

*#include* <fstream>

*#include* <string>

*#include* <vector>

*using* *namespace* std;

bool isVowel(char *c*) {

*c* *=* tolower(*c*);

*return* (*c* *==* 'a' *||* *c* *==* 'e' *||* *c* *==* 'i' *||* *c* *==* 'o' *||* *c* *==* 'u');

}

int countVowels(const string& *str*) {

    int count *=* *0*;

*for* (char c : *str*) {

*if* (isVowel(c)) {

            count*++*;

        }

    }

*return* count;

}

int main() {

    ifstream inputFile("F1.txt");

    ofstream outputFile("F2.txt");

*if* (*!*inputFile.*is\_open*() *||* *!*outputFile.*is\_open*()) {

        cout *<<* "Error: Unable to open file" *<<* endl;

*return* *1*;

    }

    vector*<*string*>* lines;

    string line;

*while* (getline(inputFile, line)) {

        lines.*push\_back*(line);

    }

    int maxVowels *=* *0*;

    int maxVowelsLine *=* *0*;

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

        int vowels *=* countVowels(lines*[*i*]*);

*if* (vowels *>* maxVowels) {

            maxVowels *=* vowels;

            maxVowelsLine *=* i;

        }

    }

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

*if* (i *!=* maxVowelsLine) {

            outputFile *<<* lines*[*i*]* *<<* endl;

        }

    }

    cout *<<* "Line with most vowels: " *<<* maxVowelsLine *+* *1* *<<* endl;

    inputFile.*close*();

    outputFile.*close*();

*return* *0*;

}

Код vns lab 9

Завдання №4 Algotester 4.1 <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/778/files#diff-635c9d12541cddd74e7b49a2f1b8767b7f2df764ceaa166d658bd30d7a2c9fa0>

*#include* <iostream>

*#include* <vector>

*using* *namespace* std;

vector<int> sort(vector<int>& *name*) {

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

*for* (int j *=* i *+* *1*; j *<* *name*.*size*(); j*++*) {

*if* (*name[*i*]* *>* *name[*j*]*) {

*// Swap elements if they are in the wrong order*

                int temp *=* *name[*i*]*;

*name[*i*]* *=* *name[*j*]*;

*name[*j*]* *=* temp;

            }

        }

    }

*return* *name*;

}

vector<int> rotate(vector<int>& *arr*, int *K*) {

    int N *=* *arr*.*size*();

    vector*<*int*>* rotatedArr(N);

*for* (int i *=* *0*; i *<* N; i*++*) {

        rotatedArr*[*(i *+* N *-* (*K* *%* N)) *%* N*]* *=* *arr[*i*]*;

    }

*return* rotatedArr;

}

int main (){

int N, K;

cin *>>* N;

cin *>>* K;

int arr[N];

vector*<*int*>* arrCopy;

*for*(int i *=* *0*; i *<* N; i*++*){

    cin*>>*arr[i];

}

*for* (int i *=* *0*; i *<* N; i*++*) {

        bool isDuplicate *=* false;

*for* (int j *=* *0*; j *<* arrCopy.*size*(); j*++*) {

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

                isDuplicate *=* true;

*break*;

            }

        }

*if* (*!*isDuplicate) {

            arrCopy.*push\_back*(arr[i]);

        }

    }

    vector *<*int*>* arrUniqSort *=* sort(arrCopy);

    vector *<*int*>* result *=* rotate(arrUniqSort, K);

    cout *<<* result.*size*() *<<* endl;

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

        cout *<<* result*[*i*]* *<<* " ";

    }

*return* *0*;

}

Код algotester 4.1

Завдання №5 Програма №5 VNS Algotester 4.2 <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/778/files#diff-ef0022548f0a8b7dbe88ce41b0451967aa4fb5217a49af51975c1bbc104b2772>

*#include* <iostream>

*#include* <vector>

*#include* <algorithm>

*using* *namespace* std;

int main (){

    int N, K;

    cin *>>* N;

    cin *>>* K;

    vector*<*int*>* arr(N);

    vector*<*int*>* arrCopy;

*for*(int i *=* *0*; i *<* N; i*++*){

        cin*>>*arr[i];

    }

    arrCopy *=* arr;

    sort(arrCopy.begin(), arrCopy.end());

    arrCopy.erase(unique(arrCopy.begin(), arrCopy.end()), arrCopy.end());

    rotate(arrCopy.begin(), arrCopy.begin() *+* K *%* arrCopy.size(), arrCopy.end());

    cout *<<* arrCopy.size() *<<* endl;

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

        cout *<<* arrCopy[i] *<<* " ";

    }

*return* *0*;

}

Код algotesteer 4.2

Завдання №6 Algotester 6 <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/778/files#diff-c26511f5dac2d01061831431eb1fdc03f5159969ce8e79fac0caa4854fedabbd>

*#include* <iostream>

*#include* <vector>

*#include* <bitset>

*#include* <algorithm>

*using* *namespace* std;

const int BOARD\_SIZE *=* *8*;

enum Figure { EMPTY, PAWN, ROOK, KNIGHT, BISHOP, KING, QUEEN };

bitset*<6>* hasFigure[BOARD\_SIZE][BOARD\_SIZE];

bool canMoveInDirection(int x, int y, const string board[], int dx, int dy, char figure) {

*for* (int i *=* x *+* dx, j *=* y *+* dy; i *>=* *0* *&&* i *<* BOARD\_SIZE *&&* j *>=* *0* *&&* j *<* BOARD\_SIZE; i *+=* dx, j *+=* dy) {

*if* (board[i][j] *==* figure) {

*return* true;

        }

    }

*return* false;

}

bool canMoveInDirections(int x, int y, const string board[], int dx[], int dy[], int n, char figure) {

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

        int nx *=* x *+* dx[i];

        int ny *=* y *+* dy[i];

*if* (nx *>=* *0* *&&* nx *<* BOARD\_SIZE *&&* ny *>=* *0* *&&* ny *<* BOARD\_SIZE *&&* board[nx][ny] *==* figure) {

*return* true;

        }

    }

*return* false;

}

bool canMovePawn(int x, int y, const string board[]) {

*if* ((x *-* *1* *>=* *0* *&&* y *-* *1* *>=* *0* *&&* board[x *-* *1*][y *-* *1*] *==* 'P') *||* (x *-* *1* *>=* *0* *&&* y *+* *1* *<* BOARD\_SIZE *&&* board[x *-* *1*][y *+* *1*] *==* 'P')) {

*return* true;

    }

*return* false;

}

bool canMoveRook(int x, int y, const string board[]) {

*return* canMoveInDirection(x, y, board, *-1*, *0*, 'R') *||*

           canMoveInDirection(x, y, board, *1*, *0*, 'R') *||*

           canMoveInDirection(x, y, board, *0*, *-1*, 'R') *||*

           canMoveInDirection(x, y, board, *0*, *1*, 'R');

}

bool canMoveKnight(int x, int y, const string board[]) {

    int knight\_dx[] *=* {*-2*, *-1*, *1*, *2*, *2*, *1*, *-1*, *-2*};

    int knight\_dy[] *=* {*1*, *2*, *2*, *1*, *-1*, *-2*, *-2*, *-1*};

*return* canMoveInDirections(x, y, board, knight\_dx, knight\_dy, *8*, 'N');

}

bool canMoveBishop(int x, int y, const string board[]) {

*return* canMoveInDirection(x, y, board, *-1*, *-1*, 'B') *||*

           canMoveInDirection(x, y, board, *-1*, *1*, 'B') *||*

           canMoveInDirection(x, y, board, *1*, *-1*, 'B') *||*

           canMoveInDirection(x, y, board, *1*, *1*, 'B');

}

bool canMoveKing(int x, int y, const string board[]) {

    int king\_dx[] *=* {*-1*, *-1*, *-1*, *0*, *0*, *1*, *1*, *1*};

    int king\_dy[] *=* {*-1*, *0*, *1*, *-1*, *1*, *-1*, *0*, *1*};

*return* canMoveInDirections(x, y, board, king\_dx, king\_dy, *8*, 'K');

}

bool canMoveQueen(int x, int y, const string board[]) {

*return* canMoveInDirection(x, y, board, *-1*, *-1*, 'Q') *||*

           canMoveInDirection(x, y, board, *-1*, *1*, 'Q') *||*

           canMoveInDirection(x, y, board, *1*, *-1*, 'Q') *||*

           canMoveInDirection(x, y, board, *1*, *1*, 'Q') *||*

           canMoveInDirection(x, y, board, *-1*, *0*, 'Q') *||*

           canMoveInDirection(x, y, board, *1*, *0*, 'Q') *||*

           canMoveInDirection(x, y, board, *0*, *-1*, 'Q') *||*

           canMoveInDirection(x, y, board, *0*, *1*, 'Q');

}

string processQuery(int x, int y, const string board[]) {

    char figure *=* board[x][y];

*if* (figure *!=* 'O') {

*return* "X";

    } *else* {

        string answer;

        const string figures *=* "BKNPQR"; *// Changed the order of figures*

*for* (char figure : figures) {

*switch* (figure) {

*case* 'B':

*if* (canMoveBishop(x, y, board)) answer *+=* figure;

*break*;

*case* 'K':

*if* (canMoveKing(x, y, board)) answer *+=* figure;

*break*;

*case* 'N':

*if* (canMoveKnight(x, y, board)) answer *+=* figure;

*break*;

*case* 'P':

*if* (canMovePawn(x, y, board)) answer *+=* figure;

*break*;

*case* 'Q':

*if* (canMoveQueen(x, y, board)) answer *+=* figure;

*break*;

*case* 'R':

*if* (canMoveRook(x, y, board)) answer *+=* figure;

*break*;

*default*:

*break*;

            }

        }

*return* answer.empty() *?* "O" *:* answer;

    }

}

int main() {

    string board[BOARD\_SIZE];

*for* (int i *=* *0*; i *<* BOARD\_SIZE; i*++*) {

        cin *>>* board[i];

*for* (int j *=* *0*; j *<* BOARD\_SIZE; j*++*) {

            hasFigure[i][j] *=* (board[i][j] *!=* '.');

        }

    }

    int Q;

    cin *>>* Q;

    vector*<*pair*<*int, int*>>* queries;

*for* (int m *=* *0*; m *<* Q; m*++*) {

        int a, b;

        cin *>>* a *>>* b;

        queries.push\_back({a *-* *1*, b *-* *1*});

    }

*for* (const auto *&*query : queries) {

        int x *=* query.first;

        int y *=* query.second;

        string answer *=* processQuery(x, y, board);

        cout *<<* answer *<<* endl;

    }

*return* *0*;

}

Код algotester 6

Завдання №7 Class Practice Work <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/778/files#diff-55f9ffea24f76aa067a937128998fe9d0b32dc4057440e6ee7a33c61affad1c7>

*#include* <iostream>

*#include* <fstream>

*using* *namespace* std;

enum FileOpResult { Success, Failure };

FileOpResult write\_to\_file(const char \**name*, const char \**content*) {

    ofstream file(*name*);

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

        file *<<* *content*;

        file.*close*();

*return* FileOpResult::Success;

    } *else* {

        cout *<<* "Error: Unable to open file " *<<* *name* *<<* " for writing." *<<* endl;

*return* FileOpResult::Failure;

    }

}

FileOpResult copy\_file(const char \**file\_from*, const char \**file\_to*) {

    ifstream MyReadFile(*file\_from*);

    ofstream MyCopyFile(*file\_to*);

    string mytext;

*if* (MyReadFile.*is\_open*() *&&* MyCopyFile.*is\_open*()) {

*while* (getline(MyReadFile, mytext)) {

            MyCopyFile *<<* mytext;

        }

        MyReadFile.*close*();

        MyCopyFile.*close*();

*return* FileOpResult::Success;

    } *else* {

        cout *<<* "Error: Unable to open files for copying." *<<* endl;

*return* FileOpResult::Failure;

    }

}

int main() {

    const char *\**filename *=* "file.txt";

    const char *\**file\_content *=* "Hello from ai 13, have a good day";

    FileOpResult result *=* write\_to\_file(filename, file\_content);

*if* (result *==* FileOpResult::Success) {

        cout *<<* "File write successful." *<<* endl;

    } *else* {

        cout *<<* "File write failed." *<<* endl;

*return* *1*; *// Вихід з програми через помилку запису в файл*

    }

    const char *\**file\_from *=* "copyfile.txt";

    const char *\**file\_to *=* "copyfile\_copy.txt";

    string copytext;

    ofstream copyfile ("copyfile.txt");

*if* (*!*copyfile.*is\_open*() *||* cin.*fail*()) {

        cout *<<* "Error: Unable to open copyfile.txt for writing." *<<* endl;

*return* *1*; *// Вихід з програми через помилку відкриття файлу для запису*

    }

    cout *<<* "Enter text to copy" *<<* endl;

    cin *>>* copytext;

    copyfile *<<* copytext;

    copyfile.*close*(); *// Закриття файлу після запису*

    FileOpResult copyResult *=* copy\_file(file\_from, file\_to);

*if* (copyResult *==* FileOpResult::Success) {

        cout *<<* "File copy successful." *<<* endl;

    } *else* {

        cout *<<* "File copy failed." *<<* endl;

*return* *1*; *// Вихід з програми через помилку копіювання файлу*

    }

*return* *0*;

}

Код Class Practice Work

## **5. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 VNS lab 6



Рисунок

Час затрачений на виконання завдання: 1 година

Завдання №1 VNS lab 8

Зображення, що містить текст, знімок екрана, Шрифт, дизайн

Автоматично згенерований опис

Рисунок 3

Час затрачений на виконання завдання: 2 години

Завдання №1 VNS lab 9



Рисунок 4

Час затрачений на виконання завдання: 1 година

Завдання №1 Algotester 4.1

Зображення, що містить Шрифт, текст, знімок екрана, типографія

Автоматично згенерований опис

Рисунок 5

Час затрачений на виконання завдання: 30 хвилин

Завдання №1 Algotester 4.2

Зображення, що містить Шрифт, знімок екрана, число, типографія

Автоматично згенерований опис

Рисунок 6

Час затрачений на виконання завдання: 30 хвилин

Завдання №1 Algotester 6

Зображення, що містить текст, знімок екрана, Шрифт, дизайн

Автоматично згенерований опис

Рисунок 7

Час затрачений на виконання завдання: 4 години

Завдання №1 Class practicity

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

Рисунок 2

Час затрачений на виконання завдання: 1 година

## **6. Кооперація з командою:**

Зображення, що містить текст, Обличчя людини, знімок екрана, відео

Автоматично згенерований опис

# **Висновки:**

Виконана робота дала ширше розуміння роботи з файлами та масивами. Також роботу з різними типами даних які використовують файли. Редагування бінарних та текстових файлів.