Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт №5**

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

***з розділу***: «Файли. Системи числення. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек»

***Виконала:***

студентка групи ШІ-13

Фесенко Дар’я Максимівна

# **Тема роботи:** Прості структури даних. Одновимірні масиви. Двовимірні Масиви. Алгоритми обробки.

**Мета роботи:** Створити програму з використанням простих структур даних, одновимірних масивів, двовимірних масивів.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: «Рядки.»
* Тема №2: «Блоковий ввід-вивід.»
* Тема №3: «Файли. Системи числення.»

1. Індивідуальний план опрацювання теорії:

* Тема №1: "Рядки".
  + Джерела Інформації
    - https://dspace.wunu.edu.ua/bitstream/316497/24160/1/fkit\_kn\_pzs\_asd\_LEK.pdf
    - https://financial.lnu.edu.ua/wp-content/uploads/2016/09/lektsiia\_VF\_4\_tema-4\_perehliad.pdf
  + Що опрацьовано:
    - Вивчено символьні і рядкові змінні і способи їхньої обробки.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 01.12.2023
  + Звершення опрацювання теми: 01.12.2023
* Тема №2: "Блоковий ввід-вивід"
  + Джерела Інформації:
    - <https://cherto4ka.xyz/2020/01/16/massive_begin/>
    - https://disted.edu.vn.ua/courses/learn/12120
  + Що опрацьовано:
    - Методи блокового вводу та виводу.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 03.12.2023
  + Звершення опрацювання теми: 05.12.2023
* Тема №3: «Файли. Системи числення.»
  + Джерела Інформації:
    - https://ela.kpi.ua/bitstream/123456789/57429/1/RTF\_Informatyka1\_kurs\_lektsii.pdf
  + Що опрацьовано:
    - Робота з файлами, системами числень.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 04.12.2023
  + Звершення опрацювання теми: 05.12.2023

# **Виконання роботи:**

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

Завдання №1 Вивчення символьних і рядкових змінних і способів їхньої обробки в мові С.

**Деталі завдання:**

Задано рядок, що складається із символів. Символи поєднуються в слова.

Слова одне від одного відокремлюються одним або декількома пробілами.

Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів.

Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку

рядка у відповідності зі своїм варіантом. Для рядка знайти кількість слів у ньому.

Завдання №2 Робота із двійковими файлами, організація вводу-виводу структурованої інформації і її зберігання на зовнішніх носіях.

**Деталі завдання:** Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вмістиме, виконати знищення й додавання елементів у відповідності зі своїм варіантом, використовуючи для пошуку елементів що знищуються чи додаються, функцію. Формування, друк, додавання й знищення елементів оформити у вигляді функцій. Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.

Структура "Студент":

- прізвище, ім'я, по батькові;

- номер телефону;

- група;

- оцінки по 3 основних предметах.

Знищити всі елементи із групи із зазначеним номером, у яких середнє арифметичне оцінок менше заданого, додати елемент після елемента із заданим прізвищем.

Завдання №3 Робота з текстовими файлами, ввід-вивід текстової інформації і її зберігання на зовнішніх носіях.

**Деталі завдання:** Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію. Виконати завдання.

1) Скопіювати з файлу F1 у файл F2 всі рядки, у яких міститься не менш двох однакових слів.

2) Визначити номер слова, у якому найбільше цифр.

Завдання №4 Algotester Lab 4.

**Деталі завдання:** Вам дано масив a з N цілих чисел. Спочатку видаліть масиву a усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4]. Після цього оберніть посортовану версію масиву a на K , тобто при K = 3 масив [1, 2, 3, 4, 5, 6, 7] перетвориться на [4, 5, 6, 7, 1, 2, 3]. Виведіть результат. Input У першому рядку цілі числа N та K У другому рядку N цілих чисел - елементи масиву a Output У першому рядку ціле число N - розмір множини a У наступному рядку N цілих чисел - множина a.

**Важливі деталі для врахування в імплементації програми:**

Constraints: 1 ≤ N , K ≤ 1000 0 ≤ a i ≤ 100

Завдання №5 Algotester Lab 6.

## **Деталі завдання:** У вас є шахова дошка розміром 8 × 8 та дуже багато фігур. Кожна клітинка може мати таке значення: Пуста клітинка O Пішак P Тура R Кінь N Слон B Король K Королева Q Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1). Далі йдуть Q запитів з координатами клітинки { x , y } . На кожен запит ви маєте вивести стрічку s i - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз). У випадку, якщо на клітинці стоїть якась фігура - виведіть символ X . У випадку, якщо клітинку не атакують - виведіть O . Наявніть фігури у певній клітинці не блокує атаку для іншої фігури. Тобто якщо між турою та клітинкою стоїть інша фігура - вважається що тура атакує цю клітинку.

## Input У перших 8 рядках стрічка r o w i - стан i -го рядка дошки. У наступному рядку ціле число Q - кількість записів У наступних Q рядках 2 цілих числа x та y - координати клітинки

## Output Q разів відповідь у наступному форматі: Строка r e s u l t - усі фігури, які атакують клітинку з запиту.

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

Завдання №6 Class Practice Work.

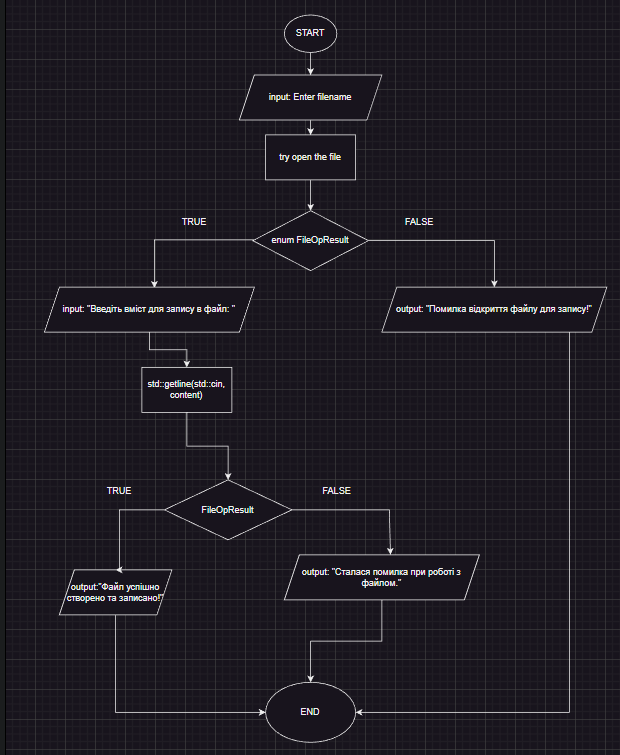


Рисунок 1 Блок-схема до програми #6

## **3. Конфігурація середовища до виконання завдань:**

Завдання №1. Конфігурація середовища

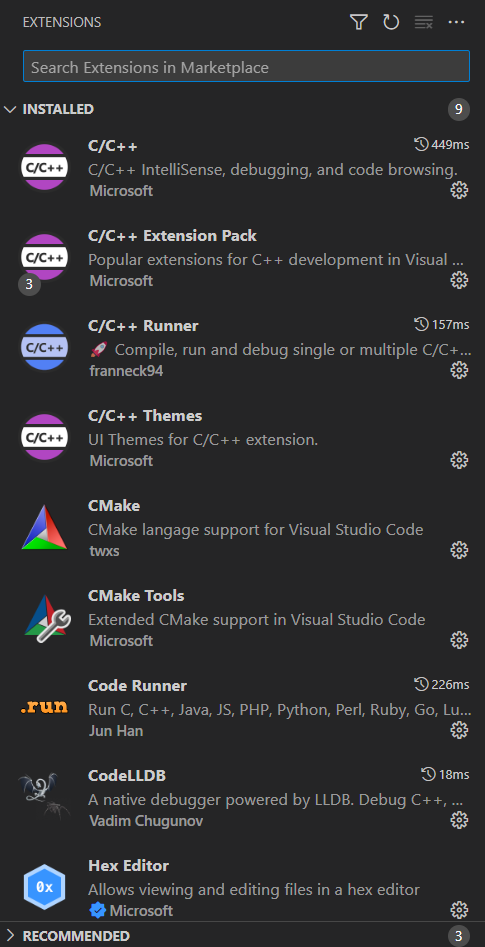
. 

Рисунок 2 Встановлені розширення

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

Завдання №1: Вивчення символьних і рядкових змінних і способів їхньої обробки в мові С.

#include <stdio.h>

#include <string.h>

int main () {

    char sentence[30];

    int words;

    printf ("Введіть речення ");

    fgets(sentence, sizeof(sentence), stdin);

    for (int s =0; s < strlen(sentence); s++ ){

        if (sentence [s] != ' ' && (sentence[s+1] == ' ' || sentence[s+1] == '\0')) {

               words++;

        }

    }

printf ("Кількість слів в рядку %d", words);

return 0;

}

Завдання №2 Робота із двійковими файлами, організація вводу-виводу структурованої інформації і її зберігання на зовнішніх носіях.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct {

    char surname[40];

    char name[40];

    char father\_name[40];

    char phone\_number[20];

    char group[10];

    float grades[3];

} STUDENT;

void addStudentToFile(FILE \*file) {

    STUDENT student;

    printf("Прізвище: "); scanf("%s", student.surname);

    printf("Ім'я: "); scanf("%s", student.name);

    printf("По батькові: "); scanf("%s", student.father\_name);

    printf("Номер телефону: "); scanf("%s", student.phone\_number);

    printf("Група: "); scanf("%s", student.group);

    printf("Оцінки (3): "); scanf("%f %f %f", &student.grades[0], &student.grades[1], &student.grades[2]);

    fwrite(&student, sizeof(STUDENT), 1, file);

}

void printStudentsFromFile(FILE \*file) {

    STUDENT student;

    while (fread(&student, sizeof(STUDENT), 1, file) == 1) {

        printf("Прізвище: %s\n", student.surname);

        printf("Ім'я: %s\n", student.name);

        printf("По батькові: %s\n", student.father\_name);

        printf("Номер телефону: %s\n", student.phone\_number);

        printf("Група: %s\n", student.group);

        printf("Оцінки: %f %f %f\n", student.grades[0], student.grades[1], student.grades[2]);

        printf("                      \n");

    }

}

void destroyStudentsInGroup(FILE \*sourceFile, FILE \*tempFile, const char groupToDelete[10], float averageGradeThreshold) {

    STUDENT student;

    rewind(sourceFile);

    rewind(tempFile);

    while (fread(&student, sizeof(STUDENT), 1, sourceFile) == 1) {

        float averageGrade = (student.grades[0] + student.grades[1] + student.grades[2]) / 3.0;

        if (strcmp(student.group, groupToDelete) != 0 || averageGrade >= averageGradeThreshold) {

            fwrite(&student, sizeof(STUDENT), 1, tempFile);

        }

    }

    rewind(tempFile);

    rewind(sourceFile);

    while (fread(&student, sizeof(STUDENT), 1, tempFile) == 1) {

        fwrite(&student, sizeof(STUDENT), 1, sourceFile);

    }

}

int main() {

    FILE \*file, \*tempFile;

    if ((file = fopen("students.dat", "ab+")) == NULL) {

        perror("Помилка при відкритті файлу");

        exit(1);

    }

    if ((tempFile = fopen("temp.dat", "w+b")) == NULL) {

        perror("Помилка при відкритті тимчасового файлу");

        fclose(file);

        exit(1);

    }

    for (int i = 1; i <= 2; i++) {

        addStudentToFile(file);

    }

    printf("Студенти в файлі:\n");

    printStudentsFromFile(file);

    destroyStudentsInGroup(file, tempFile, "IT-101", 60.0);

    addStudentToFile(file);

    printf("\nСписок студентів оновлено:\n");

    printStudentsFromFile(file);

    fclose(file);

    fclose(tempFile);

    return 0;

}

Завдання №3: Робота з текстовими файлами, ввід-вивід текстової інформації і її

зберігання на зовнішніх носіях.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

#include <errno.h>

#define BUFSIZE 512

#define BUFWORD 64

static int count\_digits(const char \*word);

static int first\_word(const char \*\*s, char \*w);

static void copy\_lines\_with\_duplicate\_words(FILE \*input, FILE \*output);

static int find\_word\_with\_max\_digits(FILE \*fo);

static int is\_word(const char \*s, const char \*w);

int main()

{

    FILE \*fo;

    FILE \*fout;

    char line[BUFSIZE];

    if ((fo = fopen("C:/Users/Acer/Desktop/F1.txt", "rt")) == NULL)

    {

        printf("Помилка відкриття вхідного файлу!!!\n");

        return 2;

    }

    if ((fout = fopen("C:/Users/Acer/Desktop/F2.txt", "wt+")) == NULL)

    {

        printf("Помилка відкриття вихідного файлу!!!\n");

        fclose(fo);

        return 3;

    }

    copy\_lines\_with\_duplicate\_words(fo, fout);

    fclose(fo);

    fclose(fout);

    if ((fo = fopen("C:/Users/Acer/Desktop/F2.txt", "rt")) == NULL)

    {

        printf("Помилка відкриття файлу F2!!!\n");

        return 4;

    }

    int max\_digits\_word\_number = find\_word\_with\_max\_digits(fo);

    printf("Слово з максимальною кількістю цифр знаходиться на позиції: %d\n", max\_digits\_word\_number);

    fclose(fo);

    return 0;

}

static void copy\_lines\_with\_duplicate\_words(FILE \*input, FILE \*output)

{

    char line[BUFSIZE];

    char word[BUFWORD];

    const char \*ptr;

    while ((ptr = fgets(line, sizeof(line), input)) != NULL)

    {

        const char \*temp\_ptr = ptr;

        while (first\_word(&temp\_ptr, word))

        {

            if (is\_word(line, word))

            {

                fputs(line, output);

                break;

            }

        }

    }

}

static int count\_digits(const char \*word)

{

    int count = 0;

    while (\*word)

    {

        if (isdigit(\*word))

        {

            count++;

        }

        word++;

    }

    return count;

}

static int find\_word\_with\_max\_digits(FILE \*fo)

{

    int max\_digits = 0;

    int current\_word\_number = 0;

    int max\_digits\_word\_number = 0;

    char line[BUFSIZE];

    const char \*ptr;

    while ((ptr = fgets(line, sizeof(line), fo)) != NULL)

    {

        const char \*temp\_ptr = ptr;

        while (first\_word(&temp\_ptr, line))

        {

            char word[BUFWORD];

            int digits = count\_digits(line);

            if (digits > max\_digits)

            {

                max\_digits = digits;

                max\_digits\_word\_number = current\_word\_number;

            }

            current\_word\_number++;

        }

    }

    return max\_digits\_word\_number;

}

static int first\_word(const char \*\*s, char \*w)

{

    const char \*p = \*s;

    char \*t = w;

    while (\*p && !isalpha(\*p))

        ++p;

    while (isalpha(\*p) && (w - t < BUFWORD))

        \*w++ = \*p++;

    \*w = '\0';

    \*s = p;

    return (w != t);

}

static int is\_word(const char \*s, const char \*w)

{

    const char \*t = s;

    const size\_t n = strlen(w);

    while ((s = strstr(s, w)) != NULL)

    {

        if ((t == s || !isalpha(\*(s - 1))) && !isalpha(\*(s + n)))

            return 1;

        s += n;

    }

    return 0;

}

Завдання №4 Algotester Lab 4.

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

    int N, K;

    cin >> N >> K;

    vector<int> a(N);

    for (int i = 0; i < N; ++i) {

        cin >> a[i];

    }

    sort(a.begin(), a.end());

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

    sort(a.begin(), a.end());

    K = K % a.size();

    rotate(a.begin(), a.begin() + K, a.end());

    cout << a.size() << endl;

    for (const auto& element : a) {

        cout << element << " ";

    }

    return 0;

}

Завдання №5 Algotester Lab 6.

#include <iostream>

#include <vector>

#include <bitset>

#include <algorithm>

using namespace std;

const int boardSize = 8;

enum Figure { empty, rook, pawn, knight, bishop, king, queen };

bitset<6> hasFigure[boardSize][boardSize];

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

    int i = x + dx;

    int j = y + dy;

    while (i >= 0 && i < boardSize && j >= 0 && j < boardSize) {

        if (board[i][j] == figure) {

            return true;

        }

        i += dx;

        j += dy;

    }

    return false;

}

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

    int i = 0;

    while (i < n) {

        int nx = x + dx[i];

        int ny = y + dy[i];

        if (nx >= 0 && nx < boardSize && ny >= 0 && ny < boardSize && board[nx][ny] == figure) {

            return true;

        }

        i++;

    }

    return false;

}

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

    return (x - 1 >= 0 && (y - 1 >= 0 && board[x - 1][y - 1] == 'P' || y + 1 < boardSize && board[x - 1][y + 1] == 'P'));

}

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

    int dx[] = {-1, 1, 0, 0};

    int dy[] = {0, 0, -1, 1};

    for (int i = 0; i < 4; ++i) {

        if (canMoveInDirection(x, y, board, dx[i], dy[i], 'R')) {

            return true;

        }

    }

    return false;

}

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};

    for (int i = 0; i < 8; ++i) {

        int nx = x + knight\_dx[i];

        int ny = y + knight\_dy[i];

        if (nx >= 0 && nx < boardSize && ny >= 0 && ny < boardSize && board[nx][ny] == 'N') {

            return true;

        }

    }

    return false;

}

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

    int dx[] = {-1, -1, 1, 1};

    int dy[] = {-1, 1, -1, 1};

    for (int i = 0; i < 4; ++i) {

        if (canMoveInDirection(x, y, board, dx[i], dy[i], 'B')) {

            return true;

        }

    }

    return false;

}

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};

    for (int i = 0; i < 8; ++i) {

        int nx = x + king\_dx[i];

        int ny = y + king\_dy[i];

        if (nx >= 0 && nx < boardSize && ny >= 0 && ny < boardSize && board[nx][ny] == 'K') {

            return true;

        }

    }

    return false;

}

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

    int dx[] = {-1, -1, -1, 0, 0, 1, 1, 1};

    int dy[] = {-1, 0, 1, -1, 1, -1, 0, 1};

    for (int i = 0; i < 8; ++i) {

        if (canMoveInDirection(x, y, board, dx[i], dy[i], 'Q')) {

            return true;

        }

    }

    return false;

}

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";

        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;

    }

}

vector<string> processQueries(const vector<string>& board, const vector<pair<int, int>>& queries) {

    vector<string> results;

    for (const auto& query : queries) {

        int x = query.first;

        int y = query.second;

        results.push\_back(processQuery(x, y, board.data()));

    }

    return results;

}

int main() {

    vector<string> board(boardSize);

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

        cin >> board[i];

    }

    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});

    }

    vector<string> results = processQueries(board, queries);

    for (const string& result : results) {

        cout << result << endl;

    }

    return 0;

}

Завдання №6 Class Practice Work.

#include <iostream>

#include <fstream>

#include <string>

enum FileOpResult { Success, Failure };

FileOpResult write\_to\_file(const std::string &name, const std::string &content) {

    std::ofstream file(name);

    if (!file.is\_open()) {

        std::cerr << "Помилка відкриття файлу для запису!" << std::endl;

        return FileOpResult::Failure;

    }

    file << content;

    file.close();

    return FileOpResult::Success;

}

int main() {

    std::string filename, content;

    std::cout << "Введіть ім'я файлу: ";

    std::getline(std::cin, filename);

    std::cout << "Введіть вміст для запису в файл: ";

    std::getline(std::cin, content);

    FileOpResult result = write\_to\_file(filename, content);

    if (result == FileOpResult::Success) {

        std::cout << "Файл успішно створено та записано!" << std::endl;

    } else {

        std::cout << "Сталася помилка при роботі з файлом." << std::endl;

    }

    return 0;

}

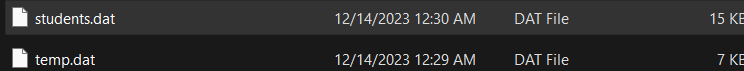
## **5. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 Вивчення символьних і рядкових змінних і способів їхньої обробки в мові С.



Рисунок 3 Результат програми 1

Завдання №2 Робота із двійковими файлами, організація вводу-виводу структурованої інформації і її зберігання на зовнішніх носіях.



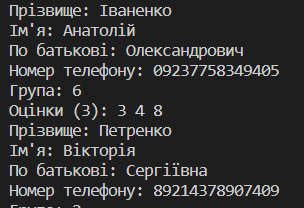


Рисунок 4 Результат програми 2

Завдання №3: Робота з текстовими файлами, ввід-вивід текстової інформації і її

зберігання на зовнішніх носіях.





Рисунок 5 Результат програми 3

Завдання №4 Algotester Lab 4.

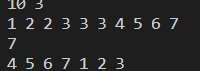


Рисунок 6 Результат програми 4

Завдання №5 Algotester Lab 6.

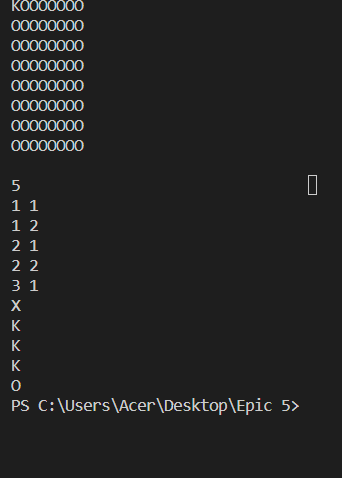


Рисунок 7 Результат програми 5

Завдання №6 Class Practice Work.

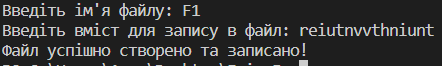


Рисунок 8 Результат програми 6

**Висновки:** Створила програму з використанням cистем числення, бінарних файлів, символів і рядкових змінних та текстових файлів, стандартних бібліотек.

**Pull request:** https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/468