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 методи/деталі роботи з файлами. Створення та використання бібліотек »

***Виконала:***

студент групи ШІ-11

Камінська Єлизавета Ігорівна

# **Тема роботи:**

Опрацювання теми роботи з бінарними файлами, методами їх використання. Дізнатись щось нове про стандартні бібліотеки.Написати коди для практичних і лабораторних робіт з ВНС, а також задач з Algotester. Залити всі дані на гітхаб і зробити pull-request.

# **Мета роботи:**

* Theory Education Activities
* Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-9)
* Lab# programming: VNS Lab 6
* Lab# programming: VNS Lab 8
* Lab# programming: VNS Lab 9
* Lab# programming: Algotester Lab 4
* Lab# programming: Algotester Lab 6
* Practice# programming: Class Practice Task
* Practice# programming: Self Practice Task
* Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)
* Results Evaluation and Release

# **Теоретичні відомості:**

1. Індивідуальний план опрацювання теорії:

* Тема №1: Theory Education Activities
  + Джерела Інформації:
    - <https://www.geeksforgeeks.org/>
    - <https://en.cppreference.com/>
    - <https://cplusplus.com/>
    - <https://www.bestprog.net/uk/2019/09/11/examples-of-using-c-tools-for-working-with-files-ua/>
    - Теорія з книжки С++ Primer 5th edition
    - Теоретичні відомості з лабораторних ВНС №4 та 5
  + Що опрацьовано:
    - Ознайомилась з роботою з бінарними файлами
    - Вивчила щось нове про бібліотеки
  + Статус: Ознайомлена
  + Початок опрацювання теми: 04.12
  + Звершення опрацювання теми: 06.12

# **Виконання роботи:**

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

Завдання №1: VNS Lab 6

* Варіант завдання: 8
* 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:

8. Перетворити рядок так, щоб всі слова в ньому стали ідентифікаторами, слова

які складаються тільки з цифр - знищити.

* Завдання №2: VNS Lab 8
* Варіант завдання: 8
* 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.

8. Структура "Покупець":

- прізвище, ім'я, по батькові;

- домашня адреса;

- номер телефону;

- номер кредитної картки.

Знищити 3 елементи з початку файлу, додати 3 елементи в кінець файлу.

* Завдання №2: VNS Lab 9
* Варіант завдання: 8
* Деталі завдання:Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію. Скопіювати з файлу F1 у файл F2 всі рядки, які не містять цифри.
* Завдання №3: Algotester Lab 4
* Варіант завдання: 2
* Деталі завдання: Вам дано масив a з N цілих чисел.Спочатку видаліть масиву a всi елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4].

Пiсля цього обернiть посортовану версiю 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.

Завдання №3: Algotester Lab 6

* Варіант завдання: 2
* Деталі завдання: У вас є шахова дошка розміром 8 × 8 та дуже багато фігур.

Кожна клітинка може мати таке значення:

• Пуста клiтинка O

• Пішак P

• Тура R

• Кiнь N

• Слон B

• Король K

• Королева Q

Вам дають позиції фігур на дошці (всі фiгури одного кольору, кількість королів може бути > 1). Далі йдуть Q запитів з координатами клітин {x, y}. На кожен запит ви маєте вивести стрічку si - 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.

Наявність фiгури у певнiй клітинці не блокує атаку для іншої фiгури. Тобто якщо між турою та клітинкою стоїть iнша фiгура - вважається що тура атакує цю клітинку.

Завдання №5: Class Practice Task

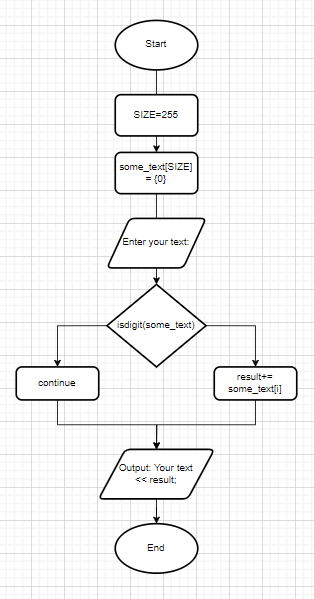
* Деталі завдання:Реалізувати функцію створення файла і запису в нього даних.
* створити файл із заданим ім’ям; якщо файл існує – перезаписати його вміст
* написати код стійкий до різних варіантів вхідних параметрів
* name – ім’я, може не включати шлях
* записати у файл вміст стрічки content, прочитати content із стандартного вводу
* повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файла.

Завдання №6: Self Practice Task

* Деталі завдання: Ця програма реалізує простий менеджер контактів за допомогою консольного інтерфейсу. Вона дозволяє користувачеві додавати нові контакти, відображати список наявних контактів та виходити з програми. Контакти зберігаються в бінарному файлі з ім'ям "contacts.dat". Кожен контакт має поля для імені, номеру телефону та електронної пошти.

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

Програма №1 : VNS Lab 6



*Блоксхема до завдання. 1: VNS Lab 6.*

* Планований час на реалізацію: 20 хв

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

Завдання №1 VNS Lab 6

#include <iostream>

#include <cstdio>

#include <string>

#include<ctype.h>

#include<stdio.h>

using namespace std;

int main(){

const int SIZE = 255;

char some\_text[SIZE] ={0};

cout << "Enter your text:";

gets\_s(some\_text, SIZE);

int i=0;

string result;

for(i=0; i< SIZE; i++){

if (isdigit(some\_text[i])){

continue;

}else {

result+= some\_text[i];

}

}

cout << "Your text:" << result;

return 0;

}

Завдання №2: VNS Lab 8

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct Buyer{

string personal\_information;

string home\_adress;

string home\_number;

string credit\_card;

};

int write(Buyer warr[]){

ofstream wf("user.bin", ios::out | ios::binary);

if(!wf){

cout <<"Cannot open the file.";

return 1;

}

for (int i=0; i<4; i++)

wf.write((char \*) &warr[i], sizeof(Buyer));

wf.close();

if(!wf.good()) {

cout << "Error occurred at writing time!" << endl;

return 1;

}

return 0;

}

int read(Buyer rarr[]){

ifstream rf("user.bin", ios::out | ios::binary);

if(!rf){

cout <<"Cannot open the file.";

return 1;

}

for(int i = 0; i < 4; i++)

rf.read((char \*) &rarr[i], sizeof(Buyer));

rf.close();

if(!rf.good()) {

cout << "Error occurred at reading time!" << endl;

return 1;

}

return 0;

}

int main() {

const int SIZE = 4;

Buyer warr[SIZE];

warr[0].personal\_information = "Kaminska Liza";

warr[0].home\_adress = "Pr.Chervonoi Kalyny 58";

warr[0].home\_number = "063-146-43-02";

warr[0].credit\_card = "123456789";

warr[1].personal\_information = "Spodaryk Sofiia";

warr[1].home\_adress = "Pidgolosko str.";

warr[1].home\_number = "098-678-56-45";

warr[1].credit\_card = "987654321";

warr[2].personal\_information = "Bokalo Yaryna";

warr[2].home\_adress = "Trylovskogo str.";

warr[2].home\_number = "076-675-87-34";

warr[2].credit\_card = "123459876";

warr[3].personal\_information = "Malanchuk Olesia";

warr[3].home\_adress = "Okruzna str.";

warr[3].home\_number = "054-567-57-43";

warr[3].credit\_card = "987612345";

write(warr);

Buyer rarr[SIZE];

read(rarr);

cout << "Buyer details:" << endl;

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

cout << "Personal information:" << rarr[i].personal\_information << endl;

cout << "Home adress:" << rarr[i].home\_adress << endl;

cout << "Home number:" << rarr[i].home\_number << endl;

cout << "Credit card:" << rarr[i].credit\_card << endl;

cout << endl;

}

Buyer rarr2[4];

rarr2[0] = rarr[3];

write(rarr2);

read(rarr2);

cout << "New buyer details:" << endl;

for(int i =0; i<4; i++){

cout << "Personal information:" << rarr2[i].personal\_information << endl;

cout << "Home adress:" << rarr2[i].home\_adress << endl;

cout << "Home number:" << rarr2[i].home\_number << endl;

cout << "Credit card:" << rarr2[i].credit\_card << endl;

cout << endl;

}

rarr2[1] = rarr[0];

rarr2[2] = rarr[1];

rarr2[3] = rarr[2];

write(rarr2);

read(rarr2);

for(int i=0; i < 4; i++) {

cout << "Personal information:" << rarr2[i].personal\_information << endl;

cout << "Home adress:" << rarr2[i].home\_adress << endl;

cout << "Home number:" << rarr2[i].home\_number << endl;

cout << "Credit card:" << rarr2[i].credit\_card << endl;

cout << endl;

}

return 0;

}

Завдання №2: VNS Lab 9

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <algorithm>

using namespace std;

void write(const string& filename,const vector<string>& data){

ofstream file(filename);

if(file.is\_open()){

for(const auto& line : data){

file << line << endl;

}

file.close();

} else{

cout << "Unable to open file for writing." << endl;

}

}

void read(const string& filename, vector<string>& data){

ifstream file(filename);

string line;

if(file.is\_open()){

while(getline(file,line)){

data.push\_back(line);

}

file.close();

}else {

cout << "Unable to open file for writing." << endl;

}

}

int main(){

vector<string> file\_1(10);

vector<string> file\_2(10);

file\_1[0]="Ghty";

file\_1[1]="657";

file\_1[2]="ABC";

file\_1[3]="123";

file\_1[4]="githn";

file\_1[5]="9080";

file\_1[6]="345";

file\_1[7]="Iouhtf";

file\_1[8]="AbC";

file\_1[9]="777";

write("file\_1.txt", file\_1);

vector<string> array\_1(10);

read("file\_1.txt", array\_1);

for (auto & element : file\_1)

{

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

if(isdigit(element[j])){

break;

}

else if (j == element.length() - 1) {

file\_2.push\_back(element);

}

}

};

for (auto & element : file\_2)

{

cout << element << " ";

}

return 0;

}

Завдання №3: Algotester Lab 4

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

int N, k;

cin >> N >> k;

vector<int> s(N);

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

cin >> s[i];

}

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

auto new\_end = unique(s.begin(), s.end());

s.erase(new\_end, s.end());

rotate(s.begin(), s.begin() + (k % s.size()), s.end());

cout << s.size() << endl;

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

cout << s[i] << " ";

}

return 0;

}

Завдання №4: Algotester Lab 6

#include <iostream>

#include <string>

#include <algorithm>

#include <vector>

#include <set>

using namespace std;

bool pawn(int x, int y, vector<string> board)

{

bool pawn = false;

if ((y > 0 && x > 0) && (board[x - 1][y - 1] == 'P' || (x > 0 && board[x - 1][y + 1] == 'P')))

{

pawn = true;

}

return pawn;

}

bool rook(int x, int y, vector<string> board)

{

bool rook = false;

for (int f = 0; f < 8; f++)

{

if (board[x][f] == 'R' || board[f][y] == 'R')

{

rook = true;

}

}

return rook;

}

bool knight(int x, int y, vector<string> board)

{

bool knight = false;

if (x > 0 && y > 1){

if (board[x + 2][y - 1] == 'N' || board[x + 2][y + 1] == 'N'|| board[x - 1][y + 2] == 'N'){

knight = true;

}

}else if(x > 0 && y > 1){

if ((x > 1 && y > 0) && (board[x - 2][y - 1] == 'N')) {

knight = true;

}

}else if(x > 1 && y > 1){

if (board[x - 2][y + 1] == 'N') {

knight = true;

}

}else if(x > 0 && y > 1){

if (board[x + 1][y - 2] == 'N') {

knight = true;

}

}else if(x > 0 && y > 1){

if (board[x + 1][y + 2] == 'N'){

knight = true;

}

}else if(x > 0 && y > 1){

if(board[x - 1][y - 2] == 'N'){

knight = true;

}

}

return knight;

}

bool bishop(int x, int y, vector<string> board)

{

bool bishop = false;

for (int f = 0; f < 8; f++)

{

if(x+f <=7 && y+f <=7){

if (board[x + f][y + f] == 'B' ||

(y >= f && board[x + f][y - f] == 'B') ||

(x >= f && board[x - f][y + f] == 'B') ||

((x >= f && y >= f) && board[x - f][y - f] == 'B'))

{

bishop = true;

}

}

}

return bishop;

}

bool king(int x, int y, vector<string> board)

{

bool king = false;

if ((board[x + 1][y + 1] == 'K' || board[x + 1][y] == 'K' ||

(y > 0 && board[x + 1][y - 1] == 'K') || board[x][y + 1] == 'K' ||

(y > 0 && board[x][y - 1] == 'K') || (x > 0 && board[x - 1][y + 1] == 'K') ||

(x > 0 && board[x - 1][y] == 'K') || ((x > 0 && y > 0) && board[x - 1][y - 1] == 'K')))

{

king = true;

}

return king;

}

bool queen(int x, int y, vector<string> board)

{

bool queen = false;

for (int u = 0; u < 8; u++)

if(x+u <=7 && y+u <=7){

if ((board[x + u][y + u] == 'Q' ||

y >= u && board[x + u][y - u] == 'Q' ||

x >= u && board[x - u][y + u] == 'Q' ||

(x >= u && y >= u) && board[x - u][y - u] == 'Q' ||

board[x][u] == 'Q' ||

board[u][y] == 'Q'))

{

queen = true;

}

}

return queen;

}

int main()

{

vector<string> board(8);

for (int i = 0; i < 8; i++)

{

getline(cin, board[i]);

}

int Q;

cin >> Q;

int \*\*array = new int \*[Q];

for (int i = 0; i < Q; i++)

{

array[i] = new int[2];

cin >> array[i][0] >> array[i][1];

}

for (int i = 0; i < Q; i++)

{

int x = array[i][0]-1;

int y = array[i][1]-1;

set<char> answer;

int k = 0;

if (board[x][y] != 'O')

{

cout << 'X' << endl;

continue;

}

if (pawn(x, y, board))

{

answer.insert('P');

k++;

}

if (rook(x, y, board))

{

answer.insert('R');

k++;

}

if (knight(x, y, board))

{

answer.insert('N');

k++;

}

if (bishop(x, y, board))

{

answer.insert('B');

k++;

}

if (king(x, y, board))

{

answer.insert('K');

k++;

}

if (queen(x, y, board))

{

answer.insert('Q');

k++;

}

if (k == 0 && board[x][y] == 'O')

{

cout << 'O' << endl;

continue;

}

for (char i : answer)

{

cout << i;

}

cout << endl;

answer.clear();

}

for (int i = 0; i < Q; i++)

{

delete[] array[i];

}

delete[] array;

return 0;

}

Завдання №5: Class Practice Task

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

enum FileOpResult { Success, Failure};

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

ofstream wf(name, ios::out | ios::binary);

if(!wf){

cout <<"Cannot open the file.";

return Failure;

}

wf.write((char \*) &content, sizeof(FileOpResult));

wf.close();

if(!wf.good()) {

cout << "Error occurred at writing time!" << endl;

return Failure;

}

return Success;

}

int main(){

char\* filename = "example.txt";

char content[256];

cout << "Enter the content to write to the file: ";

cin.getline(content,256);

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

if (result == Success) {

std::cout << "Success: File created and content written successfully."<< endl;

} else {

std::cout << "Operation failed." << endl;

}

return 0;

}

Завдання №6: Self Practice Task

#include <iostream>

#include <fstream>

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct Contact {

string name;

string phoneNumber;

string email;

};

void add\_contact(const Contact& contact, const string& filename) {

ofstream file(filename, ios::binary | ios::app);

if (file.is\_open()) {

file.write(reinterpret\_cast<const char\*>(&contact), sizeof(Contact));

cout << "Contact added successfully." << endl;

} else {

cout << "Unable to open the file."<< endl;

}

file.close();

}

void display\_contacts(const string& filename) {

ifstream file(filename, ios::binary);

if (file.is\_open()) {

Contact contact;

while (file.read(reinterpret\_cast<char\*>(&contact), sizeof(Contact))) {

cout << "Name: " << contact.name << endl;

cout << "Phone: " << contact.phoneNumber << endl;

cout << "Email: " << contact.email << endl;

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

}

} else {

cout << "Unable to open the file." << endl;

}

file.close();

}

int main() {

const string filename = "contacts.dat";

while (true) {

cout << "1. Add Contact"<< endl;

cout << "2. Display Contacts"<< endl;

cout << "3. Exit"<< endl;

cout << "Choose an option: ";

int choice;

cin >> choice;

switch (choice) {

case 1: {

Contact newContact;

cout << "Enter Name: ";

cin >> newContact.name;

cout << "Enter Phone Number: ";

cin >> newContact.phoneNumber;

cout << "Enter Email: ";

cin >> newContact.email;

add\_contact(newContact, filename);

break;

}

case 2:

display\_contacts(filename);

break;

case 3:

return 0;

default:

cout << "Invalid choice. Please try again." << endl;

}

}

return 0;

}

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

Завдання №VNS Lab 6

Enter your text:Today is 6 of December.

Your text:Today is of December.

Enter your text:Multiply 14 and 56.

Your text:Multiply and .

Час затрачений на виконання завдання: 2 год.

Завдання №2: VNS Lab 8

Buyer details:

Personal information:Kaminska Liza

Home adress:Pr.Chervonoi Kalyny 58

Home number:063-146-43-02

Credit card:123456789

Personal information:Spodaryk Sofiia

Home adress:Pidgolosko str.

Home number:098-678-56-45

Credit card:987654321

Personal information:Bokalo Yaryna

Home adress:Trylovskogo str.

Home number:076-675-87-34

Credit card:123459876

Personal information:Malanchuk Olesia

Home adress:Okruzna str.

Home number:054-567-57-43

Credit card:987612345

New buyer details:

Personal information:Malanchuk Olesia

Home adress:Okruzna str.

Home number:054-567-57-43

Credit card:987612345

Personal information:

Home adress:

Home number:

Credit card:

Personal information:

Home adress:

Home number:

Credit card:

Personal information:

Home adress:

Home number:

Credit card:

Personal information:Malanchuk Olesia

Home adress:Okruzna str.

Home number:054-567-57-43

Credit card:987612345

Personal information:Kaminska Liza

Home adress:Pr.Chervonoi Kalyny 58

Home number:063-146-43-02

Credit card:123456789

Personal information:Spodaryk Sofiia

Home adress:Pidgolosko str.

Home number:098-678-56-45

Credit card:987654321

Personal information:Bokalo Yaryna

Home adress:Trylovskogo str.

Home number:076-675-87-34

Credit card:123459876

Час затрачений на виконання завдання: 3 год.

Завдання №2: VNS Lab 9

Ghty ABC githn Iouhtf AbC

Час затрачений на виконання завдання: 2 год.

Завдання №3: Algotester Lab 4

10 3

1 2 2 3 3 3 4 5 6 7

7

4 5 6 7 1 2 3

Час затрачений на виконання завдання: 2 год

Завдання №4: Algotester Lab 6

KOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

5

1 1

1 2

2 1

2 2

3 1

X

K

K

K

O

OOOOOOOO

OQOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

13

1 1

1 2

1 3

2 1

2 2

2 3

3 1

3 2

3 3

4 4

7 7

1 7

2 7

Q

Q

Q

Q

X

Q

Q

Q

Q

Q

Q

O

Q

Час затрачений на виконання завдання: 5 год

Завдання №5: Class Practice Task

Enter the content to write to the file: Per aspera ad astra)

Success: File created and content written successfully.

Час затрачений на виконання завдання: 1.5 год

Завдання №6: Self Practice Task

1. Add Contact

2. Display Contacts

3. Exit

Choose an option: 1

Enter Name: Liza

Enter Phone Number: 087-567-56-35

Enter Email: lalala@lala.lala

Contact added successfully.

1. Add Contact

2. Display Contacts

3. Exit

Choose an option: 2

Name: Liza

Phone: 087-567-56-35

Email: lalala@lala.lala

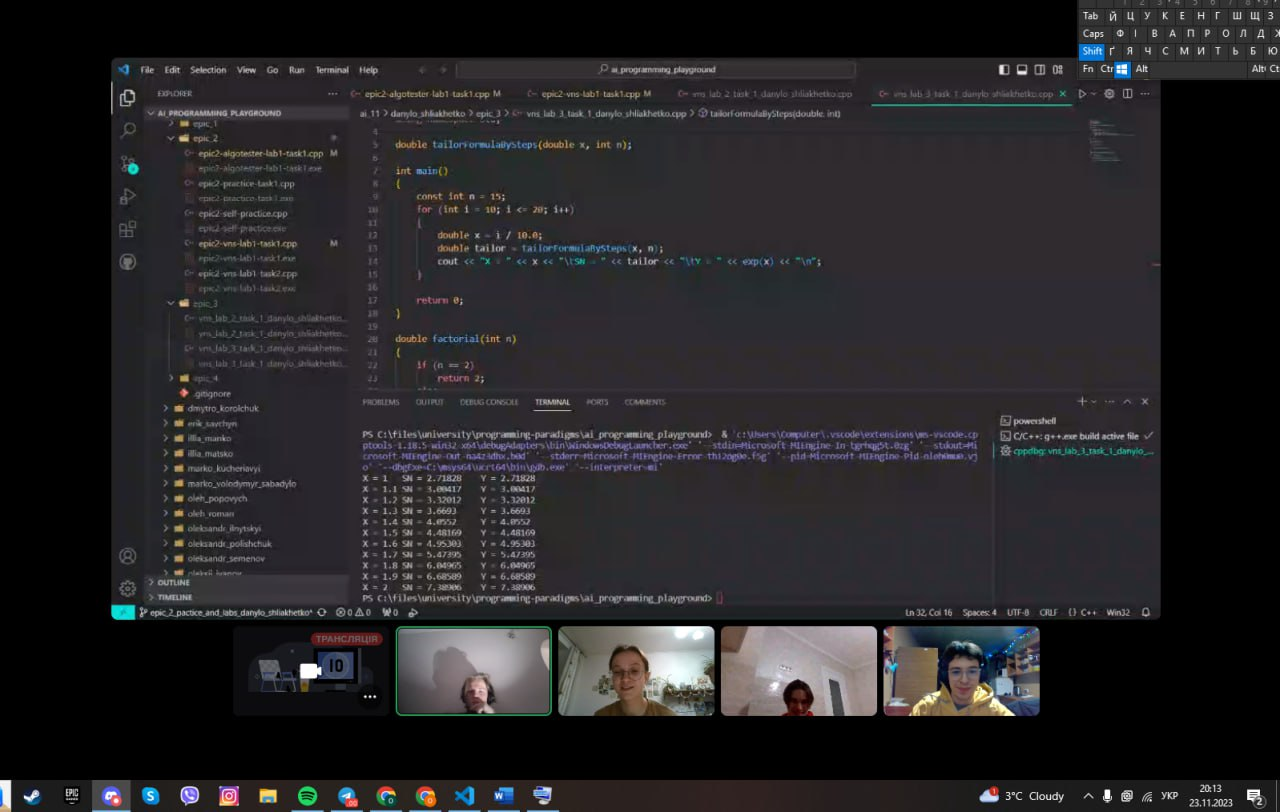
-----------------

Час затрачений на виконання завдання: 50 хв

## **5. Командна робота та комунікація:**

Наша команда цього тижня провела лише 1 зустріч, але дуже грунтовну:

На ній ми обговорили план на цей епік, завантажили таски у трело, а також обговорили питання стосовно коду.



*Зустріч у діскорді.*

# **Висновки:**

# На цій лабораторній роботі я дізналась про бінарні файли, роботу з текстовими файлами та методи їх використання . Крім цього я ознайомилась з стандартними бібліотеками та їх використанням. Щоб закріпити вивчений матеріал нам було дано декілька програм з ВНС і Algotester. Всі файли включно зі звітом запушила на гітхаб.