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»

***Виконав:***

студент групи ШІ-14

Максимів Данило Юрійович

# **Тема роботи:**

Бінарні та текстові файли. Робота з файлами у С++. Символи та рядкові змінні. Зчитування з файлу та запис у файл. Створення й використання бібліотек

# **Мета роботи:**

Ознайомитися з файлами та з деталями роботи з ними у С++. Опрацювати зчитування та запис у файл. Ознайомитися з символами та рядковими змінним

# **Теоретичні відомості:**

* 1) Теоретичні відомості з переліком важливих тем:
* - Тема №1: Файли. Бінарні та текстові файли. Використання файлів. Робота з файлами.
* - Тема №2: Стандартна бібліотека Створення й використання бібліотек»
* - Тема №3: Символи та рядкові змінні.

1. Індивідуальний план опрацювання теорії:

* Індивідуальний план опрацювання теорії:
* Тема №1: Файли. Бінарні та текстові файли. Використання файлів. Робота з файлами.
  + Джерела Інформації:
    - Лекційний матеріал ВНС
    - <https://www.w3schools.in/cplusplus/working-with-files>
  + Що опрацьовано:
    - Бінарні та текстові файли і різниця між ними.Способи запису та читання у\з файл\у,копіювання та видалення елментів з файлу
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 10.12.2023
* Тема №2: Стандартна бібліотека Створення й використання бібліотек»
  + Джерела Інформації:
    - Лекційний матеріал ВНС
    - <https://acode.com.ua/urok-205-kontejnery-stl/>
  + Що опрацьовано:
    - Методи роботи з стандартною бібліотекою.Функції **sort, unique, rotate** ,модуль **vector** і тд.
    - Зрозумів наскільки класно можна спростити собі життя
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 10.12.2023
* Тема №3: Символи та рядкові змінні.
  + Джерела Інформації:
    - Лекційний матеріал ВНС
    - безліч статей до кожної з функцій
  + Що опрацьовано:
    - Зчитування та запис рядків
    - Функції **getline,strcmp,strtok,strle**n та ін.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 10.12.2023

# **Виконання роботи:**

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

Lab# programming: VNS Lab 6



Рисунок Lab# programming: VNS Lab 6

Lab# programming: VNS Lab 8

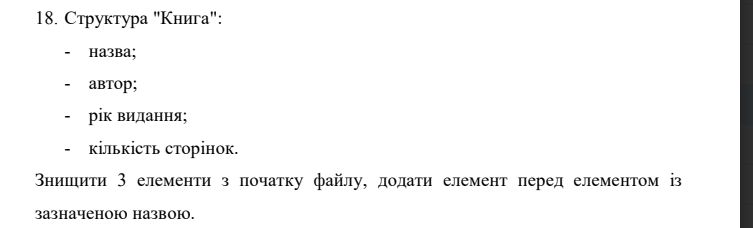


Рисунок Lab# programming: VNS Lab 8

Lab# programming: VNS Lab 9

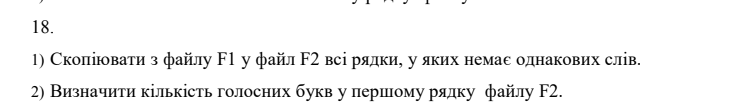


Рисунок Lab# programming: VNS Lab 9

Lab# programming: Algotester Lab 4

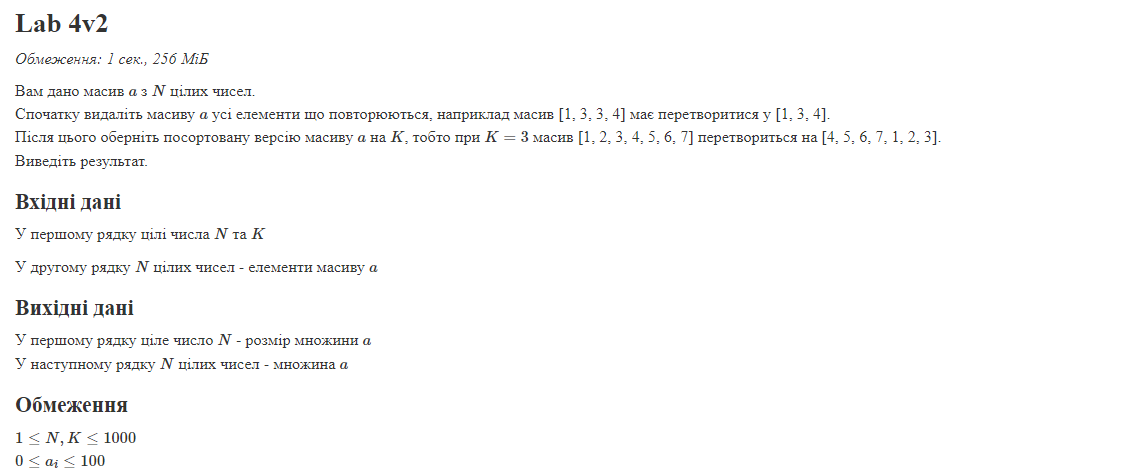


Рисунок Lab# programming: Algotester Lab 4

Lab# programming: Algotester Lab 6

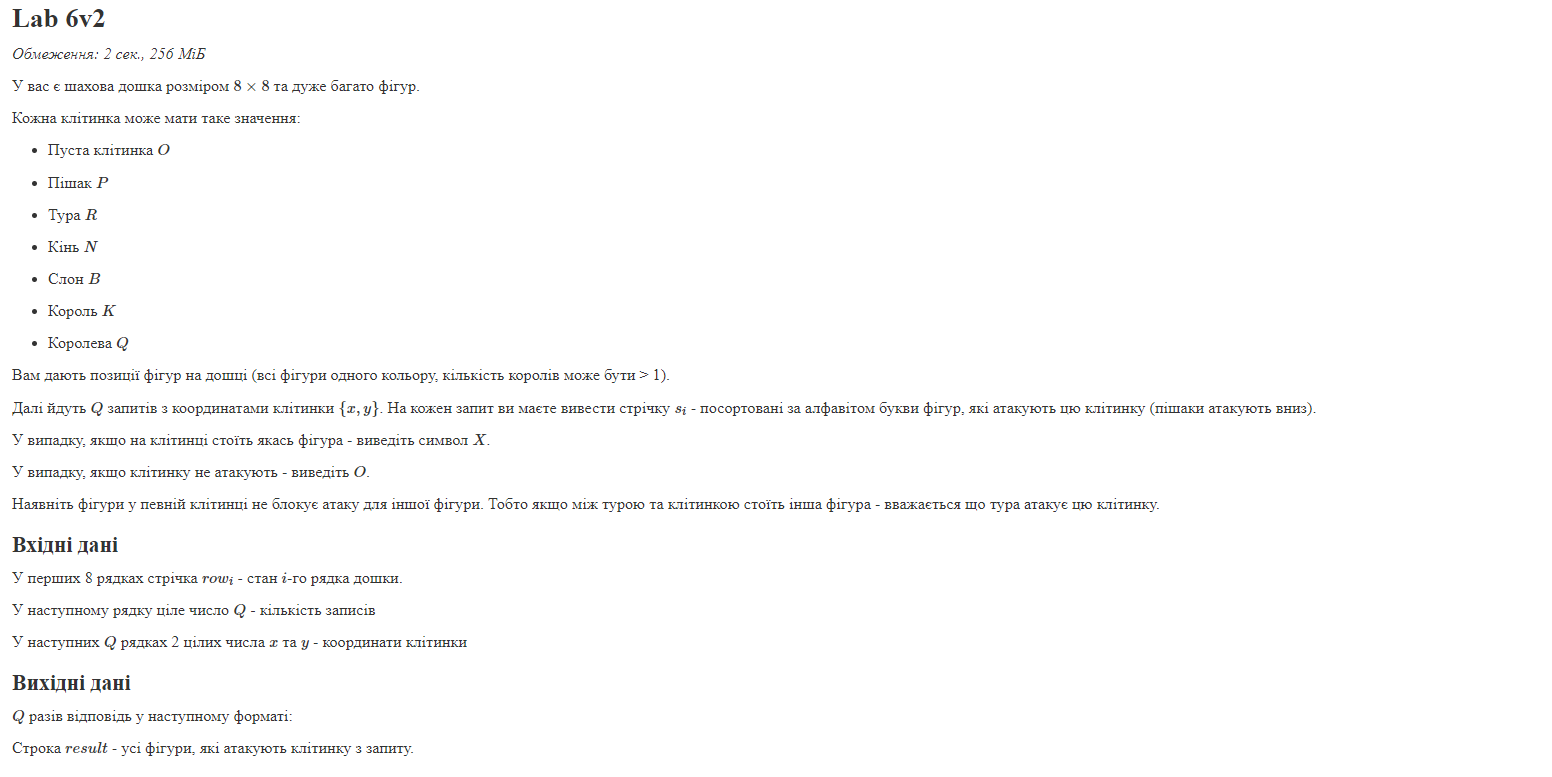


Рисунок Lab# programming: Algotester Lab 6

Practice# programming: Class Practice Task

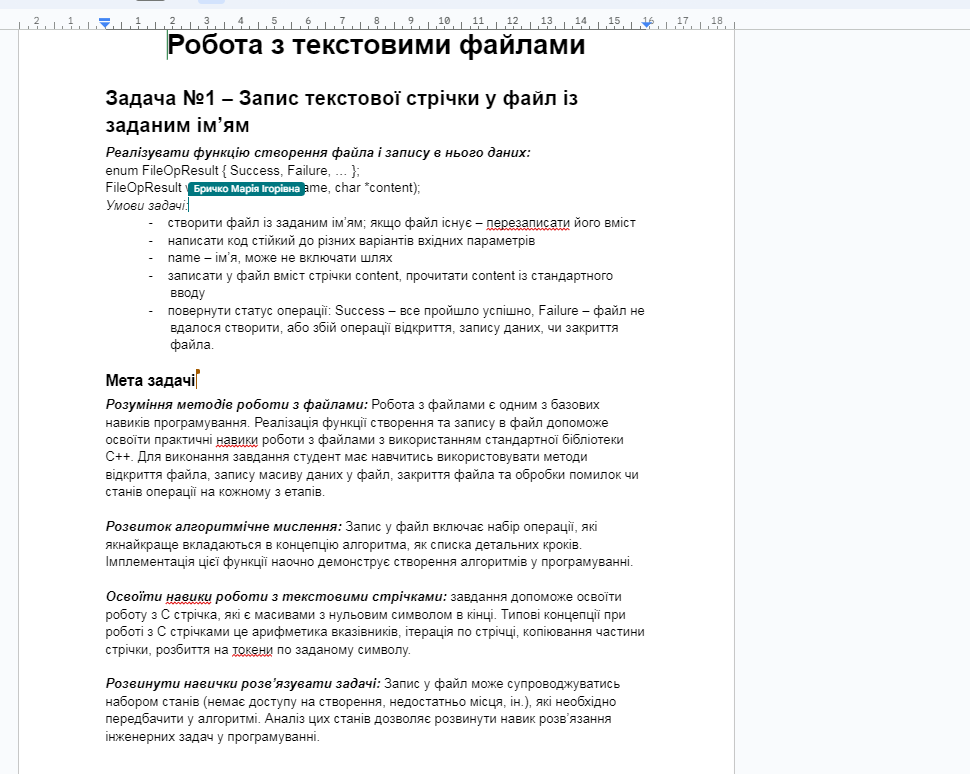


Рисунок Practice# programming: Class Practice Task

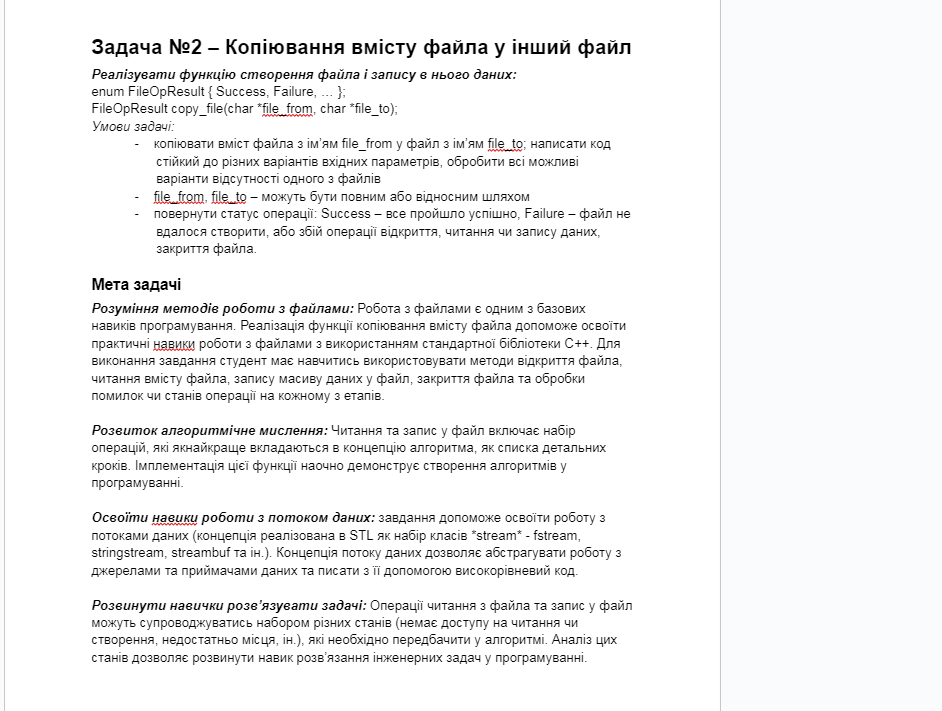


Рисунок Practice# programming: Class Practice Task

Practice# programming: Self Practice Task#1

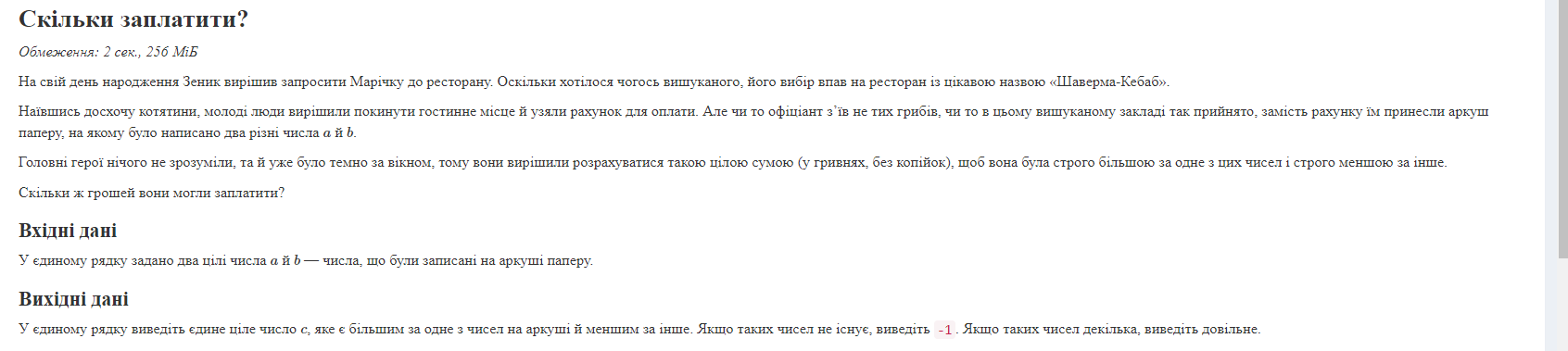


Рисунок Practice# programming: Self Practice Task#1

Practice# programming: Self Practice Task#2

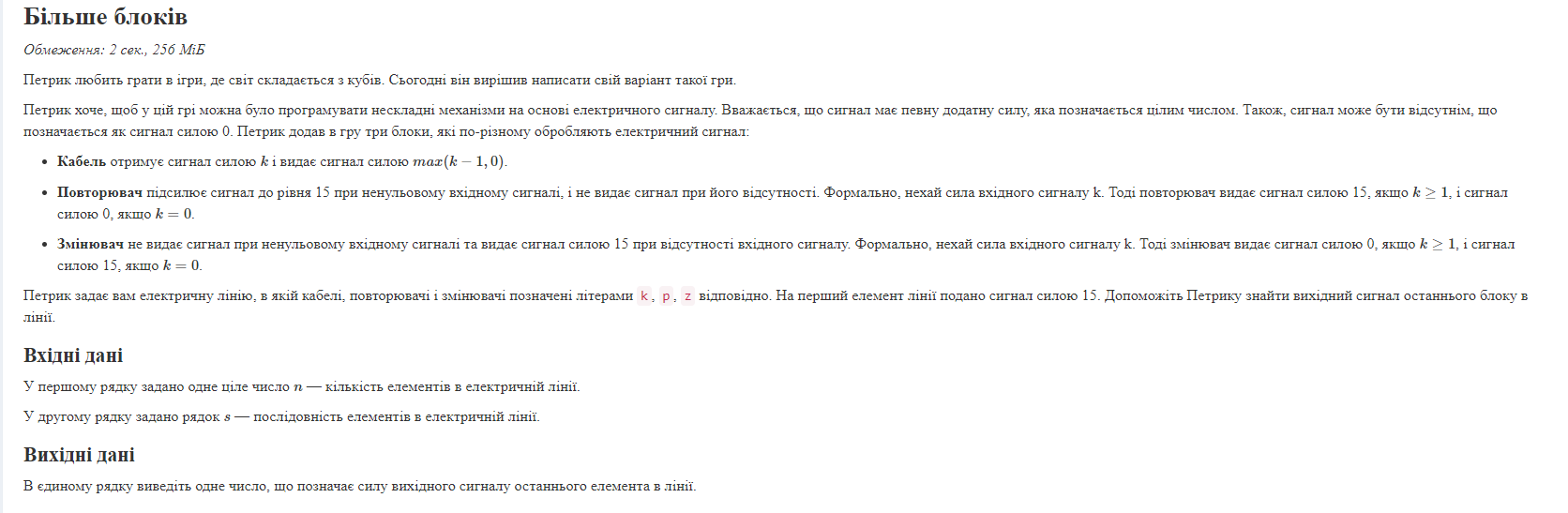


Рисунок Practice# programming: Self Practice Task#1

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

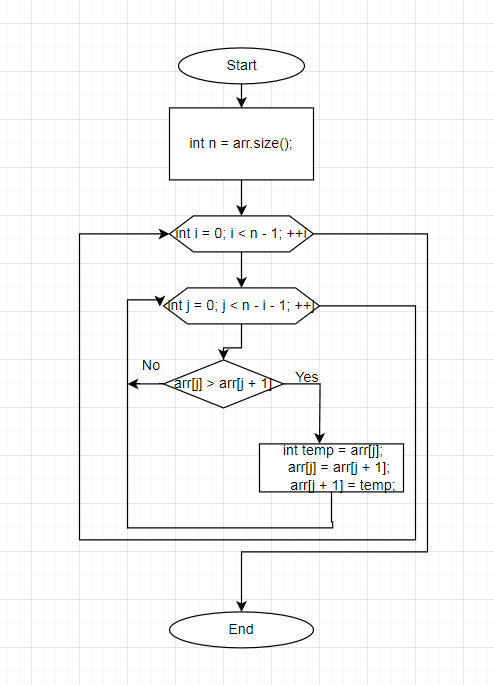


Рисунок void my\_sort

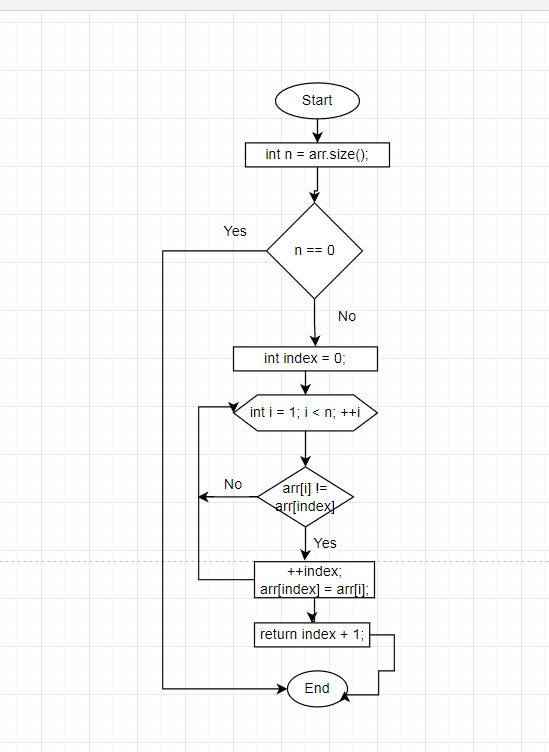


Рисунок int my\_unique

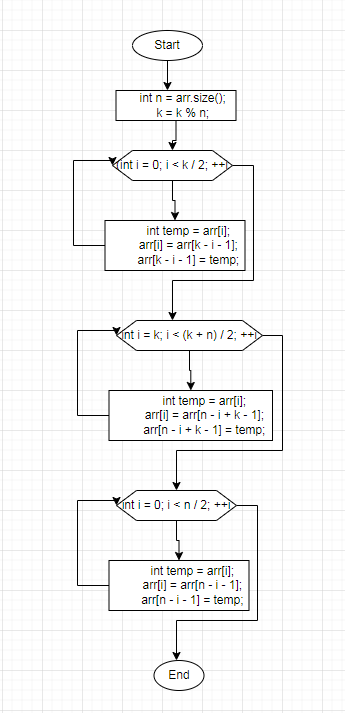


Рисунок void my\_rotate

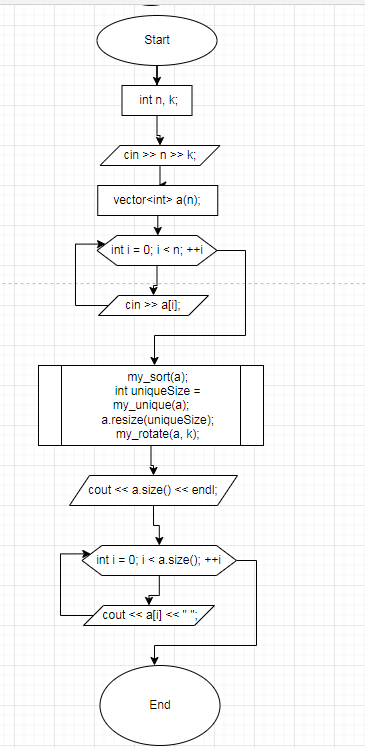


Рисунок Algotester Lab4v2

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

Середовище сконфігуровано з часу здачі епіку1

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

Lab# programming: VNS Lab 6

#include <iostream>

#include <cstring>

using namespace std;

void bubbleSortDescending(char arr[][255], int n)

{

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

{

for (int j = 0; j < n - i - 1; j++)

{

if (strcmp(arr[j], arr[j + 1]) < 0)

{

char temp[255];

strcpy(temp, arr[j]);

strcpy(arr[j], arr[j + 1]);

strcpy(arr[j + 1], temp);

}

}

}

}

int main() {

const int MAX = 255;

char str[MAX];

char strmid[MAX];

char strres[MAX][MAX];

char sortedRes[MAX][MAX];

char \*token;

int i = 0, j = 0, k = 0;

bool spaceFound = false;

cout << "\nEnter string:\n\n";

cin.getline(str, MAX, '.');

for (; str[i] != '\0'; i++)

{

if (str[i] != ' ')

{

strmid[j] = str[i];

j++;

spaceFound = false;

}

else

{

if (!spaceFound)

{

strmid[j] = ' ';

j++;

spaceFound = true;

}

}

}

strmid[j] = '\0';

i = 0;

j = 0;

token = strtok(strmid, " ");

while (token != NULL && i < MAX)

{

strcpy(strres[i], token);

token = strtok(NULL, " ");

i++;

}

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

{

if (isdigit(strres[m][0]))

{

strcpy(sortedRes[k], strres[m]);

k++;

}

}

bubbleSortDescending(sortedRes, k);

cout<<"\nSorted string:\n\n";

for (int m = 0; m < k; m++)

{

cout << sortedRes[m] << endl;

}

return 0;

}

Lab# programming: VNS Lab 8

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct Book

{

string name;

string autor;

int year;

int pages;

};

Book RichDadPoorDad = {"RichDadPoorDad", "Avanasiy", 2007, 1312};

void form(Book book)

{

ofstream f("file.bin", ios\_base::binary | ios\_base::app);

if (!f.is\_open())

{

cout << "Error opening the file." << endl;

}

else

{

f.write((char\*)(&book), sizeof(book));

f.close();

}

}

void print()

{

Book book;

ifstream f("file.bin", ios\_base::binary);

if (!f.is\_open())

{

cout << "Error opening the file." << endl;

}

else

{

while (f.read((char\*)(&book), sizeof(book)))

{

cout << book.name << ":\n";

cout << "Autor: " << book.autor << endl;

cout << "Year: " << book.year << endl;

cout << "Pages: " << book.pages << endl;

cout << endl;

}

f.close();

}

}

void del()

{

ifstream f("file.bin", ios\_base::binary);

if (!f.is\_open())

{

cout << "Error opening the file." << endl;

return;

}

ofstream tempf("tempf.bin", ios\_base::binary | ios\_base::app);

if (!tempf.is\_open())

{

cout << "Error opening the temporary file." << endl;

f.close();

return;

}

int n = 1;

Book book;

while (f.read((char\*)(&book), sizeof(book)) && n < 3)

{

n++;

}

while (f.read((char\*)(&book), sizeof(book)))

{

tempf.write((char\*)(&book), sizeof(book));

}

f.close();

tempf.close();

remove("file.bin");

rename("tempf.bin", "file.bin");

}

void add(string nametoadd)

{

ifstream f("file.bin", ios\_base::binary);

if (!f.is\_open())

{

cout << "Error opening the file." << endl;

return;

}

ofstream tempf("tempf.bin", ios\_base::binary | ios\_base::app);

if (!tempf.is\_open())

{

cout << "Error opening the temporary file." << endl;

f.close();

return;

}

Book book;

bool bookAdded = false;

while (f.read((char\*)(&book), sizeof(book)))

{

if (book.name == nametoadd)

{

tempf.write((char\*)(&RichDadPoorDad), sizeof(RichDadPoorDad));

bookAdded = true;

}

tempf.write((char\*)(&book), sizeof(book));

}

f.close();

tempf.close();

if (!bookAdded)

{

cout << "Book not found for addition." << endl;

}

remove("file.bin");

rename("tempf.bin", "file.bin");

}

int main()

{

cout << "Forming:\n\n";

Book Tora = {"Tora", "Damian", 1984, 666};

Book Meinkampf = {"Meinkampf", "Danichka", 1939, 1488};

Book It = {"It", "Vasya", 2000, 333};

Book Bible = {"Bible", "Andriy", 0, 777};

Book Koran = {"Koran", "Valodya", 1, 5};

form(Tora);

form(Meinkampf);

form(It);

form(Bible);

form(Koran);

print();

del();

cout << "After delete:\n\n";

print();

string addname;

cout << "Enter name of book before which you want to add:\n";

cin >> addname;

add(addname);

cout << "After addition:\n\n";

print();

remove("file.bin");

return 0;

}

Lab# programming: VNS Lab 9

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <sstream>

#include <unordered\_set>

using namespace std;

int eyuoia(string& s) {

int b = 0;

for (char a : s) {

if (a == 'e' || a == 'u' || a == 'i' || a == 'o' || a == 'a' || a == 'y' || a == 'E' || a == 'U' || a == 'I' || a == 'O' || a == 'A' || a == 'Y') {

b++;

}

}

return b;

}

bool isStrGood(string s) {

istringstream iss(s);

unordered\_set<string> words;

string word;

while (iss >> word) {

if (words.find(word) != words.end()) {

return false;

}

words.insert(word);

}

return true;

}

int main() {

int n, num = 0;

cout << "Enter number of strings you want to input:" << endl;

cin >> n;

cin.ignore();

vector<bool> arr;

vector<string> vec;

fstream f;

f.open("file1.txt", fstream::in | fstream::out | fstream::trunc);

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

cout << "Error opening the file." << endl;

return 1;

}

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

string msg;

cout << "Enter row #" << i + 1 << endl;

getline(cin, msg);

f << msg << "\n";

arr.push\_back(isStrGood(msg));

vec.push\_back(msg);

}

f.close();

fstream f2;

f2.open("file2.txt", fstream::in | fstream::out | fstream::trunc);

if (!f2.is\_open()) {

cout << "Error opening the file." << endl;

return 1;

}

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

if (arr[i] != 0) {

f2 << vec[i] << "\n";

}

}

f2.close();

f2.open("file2.txt", fstream::in);

if (!f2.is\_open()) {

cout << "Error opening the file." << endl;

return 1;

}

string buffer;

getline(f2, buffer);

f2.close();

int res = eyuoia(buffer);

cout << "First line from file2.txt: " << buffer << endl;

cout << "Number of vowels: " << res << endl;

return 0;

}

Lab# programming: Algotester Lab 4

#include <iostream>

#include <vector>

using namespace std;

void my\_sort(vector<int>& arr)

{

int n = arr.size();

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

{

for (int j = 0; j < n - i - 1; ++j)

{

if (arr[j] > arr[j + 1])

{

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

int my\_unique(vector<int>& arr)

{

int n = arr.size();

if (n == 0)

{

return 0;

}

int index = 0;

for (int i = 1; i < n; ++i)

{

if (arr[i] != arr[index])

{

++index;

arr[index] = arr[i];

}

}

return index + 1;

}

void my\_rotate(vector<int>& arr, int k)

{

int n = arr.size();

k = k % n;

for (int i = 0; i < k / 2; ++i)

{

int temp = arr[i];

arr[i] = arr[k - i - 1];

arr[k - i - 1] = temp;

}

for (int i = k; i < (k + n) / 2; ++i)

{

int temp = arr[i];

arr[i] = arr[n - i + k - 1];

arr[n - i + k - 1] = temp;

}

for (int i = 0; i < n / 2; ++i)

{

int temp = arr[i];

arr[i] = arr[n - i - 1];

arr[n - i - 1] = temp;

}

}

int main() {

int n, k;

cin >> n >> k;

vector<int> a(n);

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

{

cin >> a[i];

}

my\_sort(a);

int uniqueSize = my\_unique(a);

a.resize(uniqueSize);

my\_rotate(a, k);

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

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

{

cout << a[i] << " ";

}

return 0;

}

Lab# programming: Algotester Lab 4 варіант2

#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());

auto last = unique(a.begin(), a.end());

a.resize(distance(a.begin(), last));

k = k % a.size();

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

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

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

{

cout << a[i] << " ";

}

return 0;

}

Lab# programming: Algotester Lab 6

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

bool pawn (int y, int x, char board[8][8]);

bool king (int y, int x, char board[8][8]);

bool knight (int y, int x, char board[8][8]);

bool queen (int y, int x, char board[8][8]);

bool bishop (int y, int x, char board[8][8]);

bool rock (int y, int x, char board[8][8]);

int main()

{

int q;

char desk[8][8];

// ввід дошки

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

{

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

{

cin >> desk[i][j];

}

}

// ввід запитів

cin >> q;

int inputs[q][2];

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

{

for (int j = 0; j < 2; j++)

{

cin >> inputs[i][j];

}

}

// для рожного РЯДКА із запитів

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

{

int y = inputs[i][0];

int x = inputs[i][1];

x--;

y--;

vector<char> attackingFigures;

// в клітинці фігура

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

{

cout << 'X' << endl;

}

else

{

//перевірка кожної з умов

if(pawn(y,x,desk))

{

attackingFigures.push\_back('P');

}

if(rock(y,x,desk))

{

attackingFigures.push\_back('R');

}

if(queen(y,x,desk))

{

attackingFigures.push\_back('Q');

}

if(king(y,x,desk))

{

attackingFigures.push\_back('K');

}

if(knight(y,x,desk))

{

attackingFigures.push\_back('N');

}

if(bishop(y,x,desk))

{

attackingFigures.push\_back('B');

}

// клітку не атакують

if (attackingFigures.empty())

{

cout << "O";

}

else

{

// сортування і вивід вектора

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

auto last = unique(attackingFigures.begin(), attackingFigures.end());

attackingFigures.resize(distance(attackingFigures.begin(), last));

for (char a : attackingFigures)

{

cout << a;

}

}

cout<<endl;

attackingFigures.clear();

}

}

return 0;

}

bool pawn (int y, int x, char desk[8][8])

{

if (y - 1 >= 0 && x + 1 < 8 && desk[y - 1][x + 1] == 'P')

{

return true;

}

if (y - 1 >= 0 && x - 1 >= 0 && desk[y - 1][x - 1] == 'P')

{

return true;

}

return false;

}

bool rock (int y, int x, char desk[8][8])

{

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

{

if (y >= 0 && y < 8 && j >= 0 && j < 8 && (desk[y][j] == 'R' || desk[j][x] == 'R'))

{

return true;

}

}

return false;

}

bool queen (int y, int x, char desk[8][8])

{

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

{

if (y >= 0 && y < 8 && j >= 0 && j < 8 && (desk[y][j] == 'Q' || desk[j][x] == 'Q'))

{

return true;

}

}

for (int j = 1; y + j < 8 && x + j < 8; j++)

{

if (desk[y + j][x + j] == 'Q')

{

return true;

}

}

for (int j = 1; y - j >= 0 && x - j >= 0; j++)

{

if (desk[y - j][x - j] == 'Q')

{

return true;

}

}

for (int j = 1; y + j < 8 && x - j >= 0; ++j)

{

if (desk[y + j][x -j] == 'Q')

{

return true;

}

}

for (int j = 1; y - j >= 0 && x + j < 8; ++j)

{

if (desk[y - j][x + j] == 'Q')

{

return true;

}

}

return false;

}

bool bishop (int y, int x, char desk[8][8])

{

for (int j = 1; y + j < 8 && x + j < 8; ++j)

{

if (desk[y + j][x + j] == 'B')

{

return true;

}

}

for (int j = 1; y - j >= 0 && x - j >= 0; ++j)

{

if (desk[y - j][x - j] == 'B')

{

return true;

}

}

for (int j = 1; y + j < 8 && x - j >= 0; ++j)

{

if (desk[y + j][x -j] == 'B')

{

return true;

}

}

for (int j = 1; y - j >= 0 && x + j < 8; ++j)

{

if (desk[y - j][x + j] == 'B')

{

return true;

}

}

return false;

}

bool knight (int y, int x, char desk[8][8])

{

if( y+2<8 && x+1<8 && desk[y+2][x+1] == 'N' )

{

return true;

}

if( y+1<8 && x+2<8 && desk[y+1][x+2] == 'N' )

{

return true;

}

if(y-1>=0 && x+2<8 && desk[y-1][x+2] == 'N' )

{

return true;

}

if( y-2>=0 && x+1<8 && desk[y-2][x+1] == 'N' )

{

return true;

}

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

{

return true;

}

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

{

return true;

}

if(y+1<8 && x-2>=0 && desk[y+1][x-2] == 'N' )

{

return true;

}

if(y+2<8 && x-1>=0 && desk[y+2][x-1] == 'N' )

{

return true;

}

return false;

}

bool king (int y, int x, char desk[8][8])

{

if (y + 1 < 8 && x - 1 >= 0 && desk[y + 1][x - 1] == 'K')

{

return true;

}

if (y - 1 >= 0 && desk[y - 1][x] == 'K')

{

return true;

}

if (y - 1 >= 0 && x + 1 < 8 && desk[y - 1][x + 1] == 'K')

{

return true;

}

if (x + 1 < 8 && desk[y][x + 1] == 'K')

{

return true;

}

if (y + 1 < 8 && x + 1 < 8 && desk[y + 1][x + 1] == 'K')

{

return true;

}

if (y + 1 < 8 && desk[y + 1][x] == 'K')

{

return true;

}

if (y - 1 >= 0 && x - 1 >= 0 && desk[y - 1][x - 1] == 'K')

{

return true;

}

if (x - 1 >= 0 && desk[y][x - 1] == 'K')

{

return true;

}

return false;

}

Practice# programming: Class Practice Task

#include <iostream>

#include <fstream>

using namespace std;

enum FileOpResult{Success, Failure, NotExist,Failure2};

FileOpResult add(char\* name,char\* content)

{

ofstream file;

file.open(name, ios\_base::out | ios\_base::trunc);

if (!file.is\_open())

{

return Failure ;

}

file<<content;

return Success;

}

FileOpResult copy(char\* source,char\* destination)

{

const int maxContentSize = 1000;

char\* buffer= new char[maxContentSize];

ifstream file1;

ofstream file2;

file1.open (source,ios\_base::in);

if (!file1.is\_open())

{

return Failure ;

}

file2.open (destination,ios\_base::out);

if(!file1.good())

{

return NotExist;

}

if (!file2.is\_open())

{

return Failure2 ;

}

while(file1.getline(buffer,maxContentSize,'\0'))

{

file2<<buffer;

}

file1.close();

file2.close();

return Success;

}

int main()

{

FileOpResult result;

const int maxNameSize = 50, maxContentSize = 1000;

char\* name = new char[maxNameSize];

char\* content = new char[maxContentSize];

cout<<"Enter name of file or path:\n";

cin.getline(name,50);

cout<<"Enter what you want to write:\n";

cin.getline(content,1000);

result = add(name,content);

if(result == Failure)

{

cout << "Error opening the file to add something." << endl;

exit(-1);

}

else

{

cout << "File successfully opened to add something." << endl;

}

char\* source = new char[maxNameSize];

char\* destination = new char[maxNameSize];

cout<<"Enter source name of file or path:\n";

cin.getline(source,50);

cout<<"Enter destination name of file or path:\n";

cin.getline(destination,50);

result = copy(source,destination);

if(result == Failure)

{

cout << "Error opening source file." << endl;

exit(-1);

}

else if(result == Failure2)

{

cout << "Error opening destination file." << endl;

exit(-1);

}

else if(result == NotExist)

{

cout << "Source file doesn't exist." << endl;

exit(-1);

}

else

{

cout << "Copying was successful." << endl;

}

delete[]source;

delete[]destination;

delete[]name;

delete[]content;

}

Practice# programming: Self Practice Task#1

#include<iostream>

#include <cstdlib>

using namespace std;

int main()

{

long long a,b;

cin>>a>>b;

if (a-b==1 || b-a==1)

{

cout<<-1;

return 0;

}

if(a>b)

{

cout<<a-1;

}

else if(a<b)

{

cout<<b-1;

}

}

Practice# programming: Self Practice Task#2

#include <iostream>

using namespace std;

#include <string>

int main()

{

int n;

cin>>n;

string s ;

cin>>s;

int signal=15;

for(auto i = s.begin(); i != s.end(); i++)

{

if (\*i=='k')

{

signal--;

}

else if(\*i=='p')

{

if(signal>=1)

{

signal=15;

}

else if(signal==0)

{

signal=0;

}

else if(\*i=='z')

{

if(signal>=1)

{

signal=0;

}

else if(signal==0)

{

signal=15;

}

}

}

}

cout<<signal;

}

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

Lab# programming: VNS Lab 6

In 5the 3azure, 5stars and 6spring 6sings in the soul, 7Life 0is an 2incredible 1journey 4is, Find 9beauty in the little 8things 1and rejoice in 5every moment, May 1this 2day bring 4you joy and 8unforgettable 8experiences.

Sorted string:

9beauty

8unforgettable

8things

8experiences

7Life

6spring

6sings

5the

5stars

5every

4you

4is,

3azure,

2incredible

2day

1this

1journey

1and

0is

Затрачений час:6 годин

Lab# programming: VNS Lab 8

Forming:

Tora:

Autor: Damian

Year: 1984

Pages: 666

Meinkampf:

Autor: Danichka

Year: 1939

Pages: 1488

It:

Autor: Vasya

Year: 2000

Pages: 333

Bible:

Autor: Andriy

Year: 0

Pages: 777

Koran:

Autor: Valodya

Year: 1

Pages: 5

After delete:

Bible:

Autor: Andriy

Year: 0

Pages: 777

Koran:

Autor: Valodya

Year: 1

Pages: 5

Enter name of book before which you want to add:

Koran

After addition:

Bible:

Autor: Andriy

Year: 0

Pages: 777

RichDadPoorDad:

Autor: Avanasiy

Year: 2007

Pages: 1312

Koran:

Autor: Valodya

Year: 1

Pages: 5

Затрачений час 4 години

Lab# programming: VNS Lab 9

Enter number of strings you want to input:

3

Enter row #1

Hello Hello world

Enter row #2

Hi Hi llll

Enter row #3

AAAAooooooee eeeeooo

First line from file2.txt: AAAAooooooee eeeeooo

Number of vowels: 19

Затрачений час 3 години

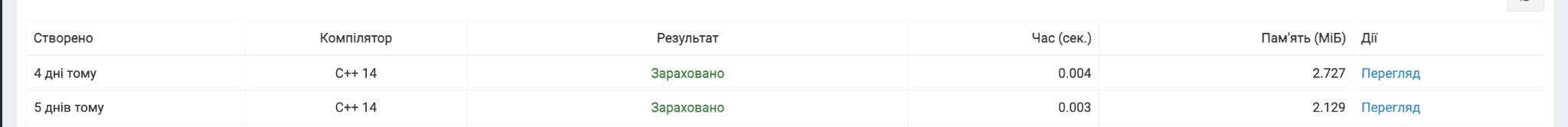
Lab# programming: Algotester Lab 4

Рисунок Lab# programming: Algotester Lab 4

Затрачений час 40 хв

Lab# programming: Algotester Lab 4 v2

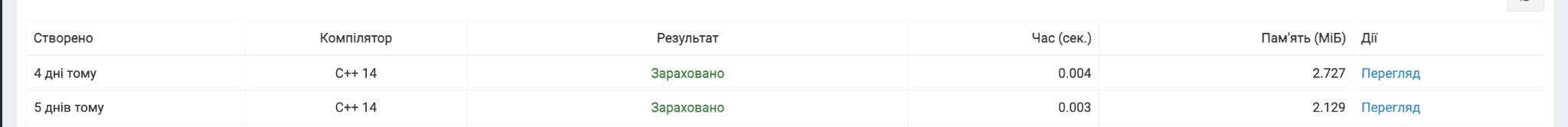


Рисунок Lab# programming: Algotester Lab 4 v2

Затрачений час 20 хв

Lab# programming: Algotester Lab 6

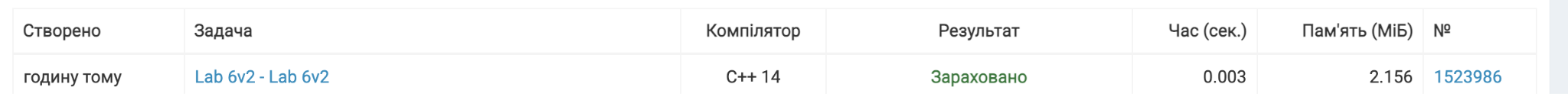


Рисунок Lab# programming: Algotester Lab 6

Затрачений час 12 годин

Practice# programming: Class Practice Task

Enter name of file or path:

test.cpp

Enter what you want to write:

SHI teachers are the best <3

File successfully opened to add something.

Enter source name of file or path:

test.cpp

Enter destination name of file or path:

test2.cpp

Copying was successful.

Затрачений час 2 години

Practice# programming: Self Practice Task#1

Рисунок Practice# programming: Self Practice Task#1

Затрачений час 20 хв

Practice# programming: Self Practice Task#2



Рисунок Practice# programming: Self Practice Task#2

Затрачений час 20 хв

## **6. Кооперація з командою:**



Рисунок chill with teammates#1🥶

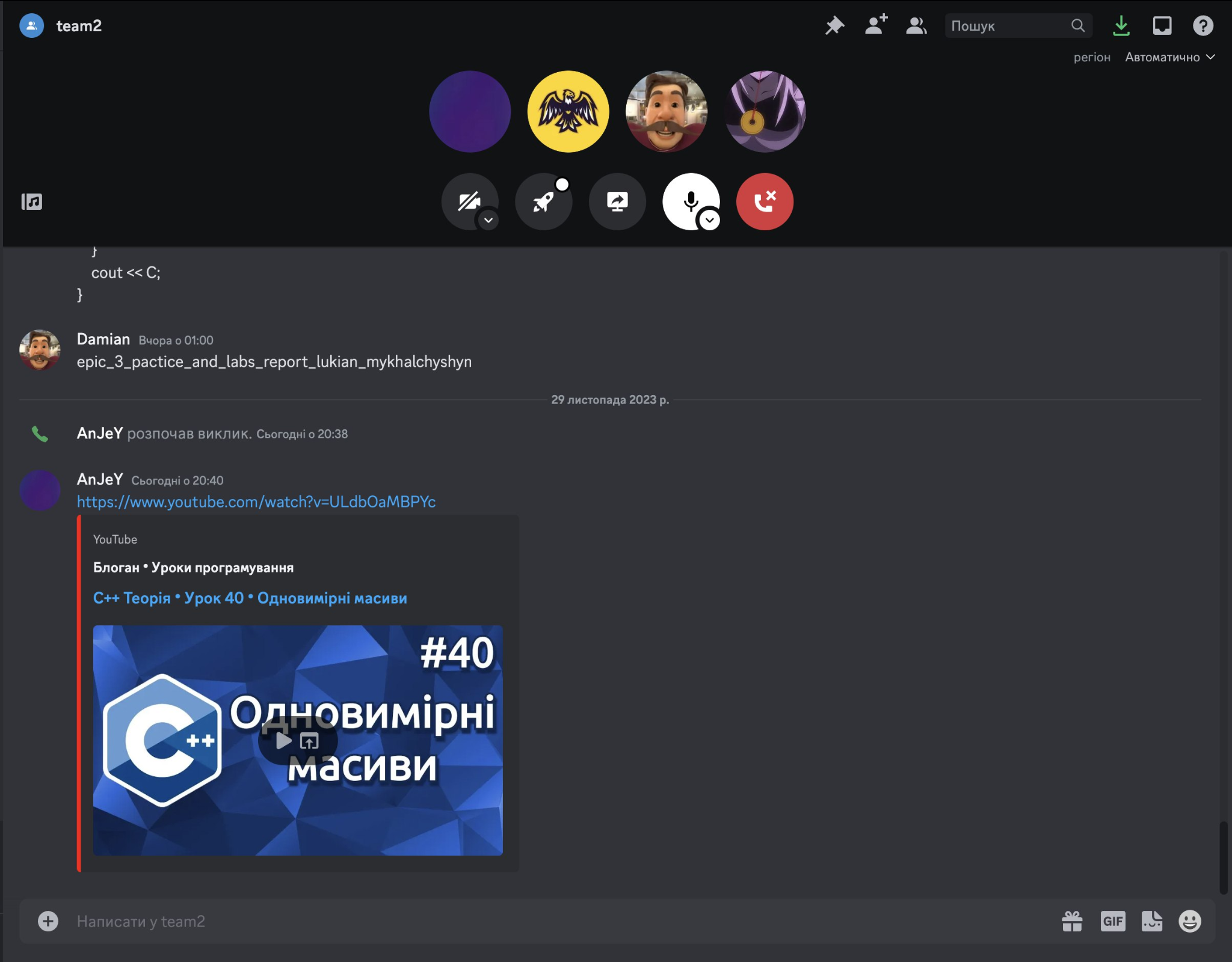


Рисунок chill with teammates#2🥶

# **Висновки:**

Ознайомився з файлами та з деталями роботи з ними у С++. Опрацював зчитування та запис у файл. Ознайомився з символами та рядковими змінним