Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 6, 8, 9**

***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 та використання бібліотек»

***Виконала:***

студентка групи ШІ-12

Борщ Анастасія Ігорівна

# **Тема роботи:**

Знання символьних і рядкових змінних мови програмування C++ і керування ними. Принцип роботи з двійковими і текстовими файлами. Ознайомлення з типовими алгоритмами роботи з файлами і бібліотеками.

# **Мета роботи:**

Навчітися використовувати символьні та рядкові змінні в C++. Вивчити основні алгоритми роботи з файлами та бібліотеками. Застосувати вивчене на практиці.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Текстові та бінарні файли.
* Тема №2: Основні методи роботи з файлами в C++.
* Тема №3: Стандартна бібліотека в C++.
* Тема №4: Робота з символьними та рядковими змінними.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Текстові та бінарні файли.
  + Джерела Інформації
    - Стаття: <https://purecodecpp.com/uk/archives/2751>
  + Що опрацьовано:
    - Опрацьовано теорію про текстові та бінарні файли, їх особливості
  + Статус: Ознайомлена
  + Початок опрацювання теми: 05.12.2023
  + Звершення опрацювання теми: 05.12.2023
* Тема №2: Основні методи роботи з файлами в C++.
  + Джерела Інформації:
    - Відео: <https://www.youtube.com/watch?v=CBnB2fvfu_I>
  + Що опрацьовано:
    - Вивчено теорію роботи з файлами та потоками в мові програмування С++
  + Статус: Ознайомлена
  + Початок опрацювання теми: 06.12.2023
  + Звершення опрацювання теми: 06.12.2023
* Тема №3: Стандартна бібліотека C++.
  + Джерела Інформації:
    - Відео: [C++ Теорія ⦁ Урок 79 ⦁ Стандартна бібліотека C++](https://www.youtube.com/watch?v=m-WJikuZGuU&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)
  + Що опрацьовано:
    - Опрацьовано поняття стандартної бібліотеки мови C++ та її використання
  + Статус: Ознайомлена
  + Початок опрацювання теми: 06.12.2023
  + Звершення опрацювання теми: 06.12.2023
* Тема №4: Робота з символьними та рядковими змінними.
  + Джерела Інформації:
    - Стаття: <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/73_C++/index.html>
    - Відео: <https://www.youtube.com/watch?v=CUTKd-O7uGM>
  + Що опрацьовано:
    - Опрацьовано основні методи роботи з символьними змінними та рядками
  + Статус: Ознайомлена
  + Початок опрацювання теми: 08.12.2023
  + Звершення опрацювання теми: 08.12.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 6

* Варіант завдання: 18
* Деталі завдання : Всі слова рядка, які починаються із цифри відсортувати за спаданням.Використання функції gets() для отримання рядка з клавіатури

Завдання №2 VNS Lab 8

* Варіант завдання: 18
* 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 "Книга":

- назва;

- автор;

- рік видання;

- кількість сторінок.

Знищити 3 елементи з початку файлу, додати елемент перед елементом із зазначеною назвою.

Завдання №3 VNS Lab 9

* Варіант завдання: 18
* Деталі завдання: Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію

1) Скопіювати з файлу F1 у файл F2 всі рядки, у яких немає однакових слів. 2) Визначити кількість голосних букв у першому рядку файлу F2.

* Важливі деталі для врахування в імплементації програми

Використання фунrції getline() для отримання стрічки з клавіатури

Завдання №4 Algotester Lab 4

* Варіант завдання: 2
* Деталі завдання: <https://algotester.com/uk/ContestProblem/DisplayWithEditor/134641>
* Важливі деталі для врахування в імплементації програми

Завдання виконано без використання засобів STL, але описано власні функції, які є аналогами функцій стандартної бібліотеки

Завдання №5 Algotester Lab 6

* Варіант завдання: 2
* Деталі завдання: <https://algotester.com/uk/ContestProblem/DisplayWithEditor/134743>
* Важливі деталі для врахування в імплементації програми

Використання власноручно написаної функції для перевірки атаки фігур

Завдання №6 Class Practice Work

* Варіант завдання: немає
* Деталі завдання: Запис текстової стрічки у файл із заданим ім’ям

Реалізувати функцію створення файла і запису в нього даних:

enum FileOpResult { Success, Failure, … };

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

Умови задачі:

- 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 №8 Self Practice Work

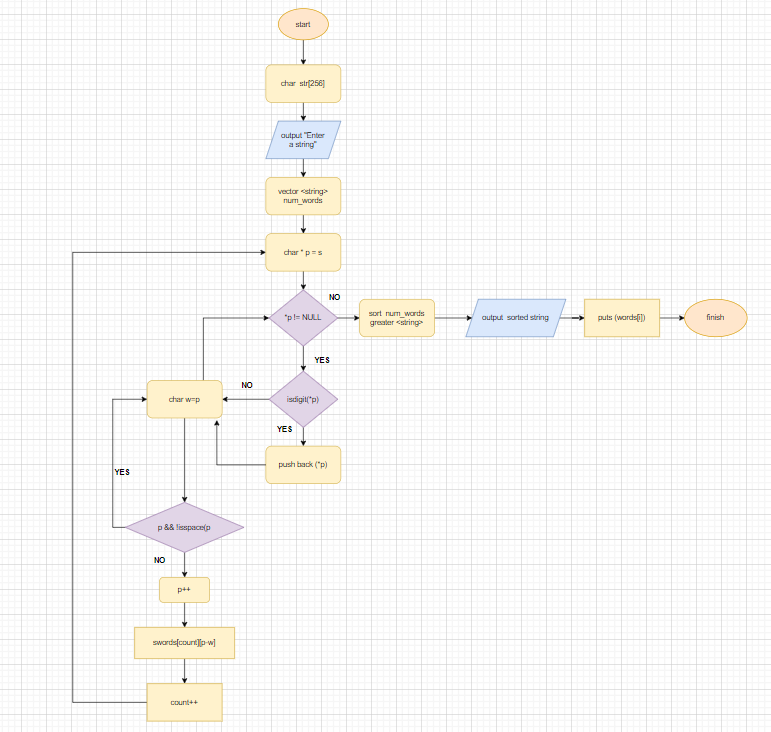
* Варіант завдання: немає
* Деталі завдання:

[**https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40120**](https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40120)

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

Програма №1 VNS Lab 6

* Блок-схема



*Рисунок 1. Блок-схема до програми №1*

* Планований час на реалізацію: 2 години

Програма №2 VNS Lab 8

* Планований час на реалізацію: 3 години

Програма №3 VNS Lab 9

* Планований час на реалізацію: 2 години

Програма №4 Algotester Lab 4

* Планований час на реалізацію: 1 година

Програма №5 Algotester Lab 6

* Планований час на реалізацію: 5 годин

Програма №6 Class Practice Work

* Планований час на реалізацію: 1 година

Програма №7 Self Practice Work

* Планований час на реалізацію: 40 хвилин

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

Для виконання поставлених завдань додаткова конфігурація середовища не є необхідною

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

Завдання №1 VNS Lab 6

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

  char s[256];

  char words[256][256];

  int count = 0;

  printf("Enter string: ");

  fgets(s, sizeof(s), stdin);

  s[strcspn(s, "\n")] = '\0';

  char \*p = s;

  while (\*p) {

    if (isdigit(\*p)) {

      char \*w = p;

      while (\*p && !isspace(\*p)) {

        p++;

      }

      strncpy(words[count], w, p - w);

      words[count][p - w] = '\0';

      count++;

    }

    p++;

  }

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

    for (int j = i + 1; j < count; j++) {

      if (strcmp(words[i], words[j]) < 0) {

        char temp[256];

        strcpy(temp, words[i]);

        strcpy(words[i], words[j]);

        strcpy(words[j], temp);

      }

    }

  }

  for (int i = 0; i < count; i++) {

    puts(words[i]);

  }

  return 0;

}

Завдання №2 VNS Lab 8

#include <iostream>

#include <vector>

#include <fstream>

#include <sstream>

using namespace std;

struct Book {

    string name;

    string author;

    int year;

    int pages;

};

void fillFile(const vector<Book>& books, const string& fileName);

void deleteBooks(vector<Book>& books, const string& fileName);

bool addBook(const string& fileName);

void printFile(const string& fileName);

int main() {

    vector<Book> books = {

        {"1984", "Orwell", 1948, 312},

        {"The Great Gatsby", "Fitzgerald", 1925, 578},

        {"War and Peace", "Leo Tolstoy", 1899, 416},

        {"Pride and Prejudice", "Jane Austen", 1813, 238},

        {"The Iliad", "Homer",1598, 160}

    };

    const string fileName = "file1.txt";

    fillFile(books, fileName);

    printFile(fileName);

    deleteBooks(books, fileName);

    if (!addBook(fileName)) return 1;

    printFile(fileName);

    return 0;

}

void fillFile(const vector<Book>& books, const string& fileName) {

    ofstream fout(fileName);

    if (!fout) {

        cerr << "Error occurred while opening the file" << endl;

        return;

    }

    for (const Book& b : books) {

        fout << b.name << "\t" << b.author << "\t" << b.year << "\t" << b.pages << endl;

    }

    fout.close();

}

void deleteBooks(vector<Book>& books, const string& fileName) {

    if (books.size() >= 3) {

        books.erase(books.begin(), books.begin() + 3);

    }

    ofstream fout("temp.bin");

    if (!fout) {

        cerr << "Error occurred while deleting first 3 books" << endl;

        return;

    }

    for (const Book& b : books) {

        fout << b.name << "\t" << b.author << "\t" << b.year << "\t" << b.pages << endl;

    }

    fout.close();

    remove(fileName.c\_str());

    rename("temp.bin", fileName.c\_str());

}

bool addBook(const string& fileName) {

    Book newbook;

    cout << "Enter the information about a book you want to add:\n";

    cout << "Title: ";

    cin.ignore();

    getline(cin, newbook.name);

    cout << "Author: ";

    getline(cin, newbook.author);

    cout << "Year: ";

    cin >> newbook.year;

    cout << "Pages: ";

    cin >> newbook.pages;

    string beforeBook;

    cout << "Before which book do you want to insert new one?\n";

    cin.ignore();

    getline(cin, beforeBook);

    ifstream fin1(fileName);

    if (!fin1) {

        cerr << "Error occurred while reading data from the file" << endl;

        return false;

    }

    ofstream fout2("temp.bin");

    if (!fout2) {

        cerr << "Error occurred while writing data into the file" << endl;

        return false;

    }

    string line;

    bool found = false;

    while (getline(fin1, line)) {

        istringstream iss(line);

        string w;

        iss >> w;

        if (w == beforeBook) {

            fout2 << newbook.name << "\t" << newbook.author << "\t" << newbook.year << "\t" << newbook.pages << endl;

            found = true;

        }

        fout2 << line << endl;

    }

    if (!found) {

        cerr << "This book doesn't exist." << endl;

        return false;

    }

    fin1.close();

    fout2.close();

    remove(fileName.c\_str());

    rename("temp.bin", fileName.c\_str());

    return true;

}

void printFile(const string& fileName) {

    ifstream fin(fileName);

    if (!fin) {

        cerr << "Error occurred while printing the file's content" << endl;

        return;

    }

    string str;

    while (getline(fin, str)) {

        cout << str << endl;

    }

    fin.close();

}

Завдання №3 VNS Lab 9

#include <iostream>

#include <fstream>

#include <sstream>

#include <vector>

#include <algorithm>

#include <cctype>

using namespace std;

string cleanWord(const string &word) {

    string cleaned;

    for (char c : word) {

        if (isalnum(c)) {

            cleaned += tolower(c);

        }

    }

    return cleaned;

}

bool isUniqueLine(const string &line) {

    istringstream iss(line);

    vector<string> lineWords;

    string word;

    while (iss >> word) {

        lineWords.push\_back(cleanWord(word));

    }

    return all\_of(lineWords.begin(), lineWords.end(), [lineWords](const string &w) {

        return count(lineWords.begin(), lineWords.end(), w) == 1;

    });

}

int countVowels(const string &str) {

    int vowels = 0;

    for (char c : str) {

        if (tolower(c) == 'a' || tolower(c) == 'e' || tolower(c) == 'i' || tolower(c) == 'o' || tolower(c) == 'u' || tolower(c) == 'y') {

            vowels++;

        }

    }

    return vowels;

}

int main() {

    string text[] = {

        "Cats purr when they're content.\n",

        "Sunflowers turn towards the sun.\n",

        "Raindrops glistened on the window pane.\n",

        "Laughter echoed through the empty hallways.\n",

        "The moonlit night was calm and serene.\n"};

    ofstream fout1("F1.txt");

    if (!fout1) {

        cerr << "Error occurred" << endl;

        return 1;

    }

    for (const string &str : text) {

        fout1 << str;

    }

    fout1.close();

    ifstream fin1("F1.txt");

    ofstream fout2("F2.txt");

    if (!fin1 || !fout2) {

        cerr << "Error occurred" << endl;

        return 1;

    }

    string line;

    while (getline(fin1, line)) {

        if (isUniqueLine(line)) {

            fout2 << line << endl;

        }

    }

    fout2.close();

    fin1.close();

    ifstream fin2("F2.txt");

    if (!fin2) {

        cerr << "Error occurred" << endl;

        return 1;

    }

    string firstLine;

    getline(fin2, firstLine);

    int vowels = countVowels(firstLine);

    fin2.close();

    cout << "There are " << vowels << " vowels in the first line\n";

    return 0;

}

Завдання №4 Algotester Lab 4

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

vector<int> mergeSort(const vector<int>& arr);

vector<int> merge(const vector<int>& left, const vector<int>& right);

vector<int> unique(const vector<int>& arr);

vector<int> rotate(const vector<int>& arr, int k);

int main() {

    int n, k;

    cin >> n >> k;

    vector<int> arr(n);

    for (int &el: arr) cin >> el;

    arr = mergeSort(arr);

    arr = unique(arr);

    arr = rotate(arr, k);

    cout << arr.size() << endl;

    for (int el: arr) cout << el << " ";

    return 0;

}

vector<int> merge(const vector<int>& left, const vector<int>& right) {

    vector<int> result;

    int i = 0, j = 0;

    while (i < left.size() && j < right.size()) {

        if (left[i] <= right[j]) {

            result.push\_back(left[i]);

            i++;

        } else {

            result.push\_back(right[j]);

            j++;

        }

    }

    while (i < left.size()) {

        result.push\_back(left[i]);

        i++;

    }

    while (j < right.size()) {

        result.push\_back(right[j]);

        j++;

    }

    return result;

}

vector<int> mergeSort(const vector<int>& arr) {

    if (arr.size() <= 1) {

        return arr;

    }

    int mid = arr.size() / 2;

    vector<int> left(arr.begin(), arr.begin() + mid);

    vector<int> right(arr.begin() + mid, arr.end());

    left = mergeSort(left);

    right = mergeSort(right);

    return merge(left, right);

}

vector<int> unique(const vector<int>& arr) {

    vector<int> result;

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

        if (i == 0 || arr[i] != arr[i - 1]) {

            result.push\_back(arr[i]);

        }

    }

    return result;

}

vector<int> rotate(const vector<int>& arr, int k) {

    k %= arr.size();

    if (k == 0) return arr;

    vector<int> result(arr.size());

    for (int i = k; i < arr.size(); i++) {

        result[i - k] = arr[i];

    }

    for (int i = 0; i < k; i++) {

        result[arr.size() - k + i] = arr[i];

    }

    return result;

}

Завдання №5 Algotester Lab 6

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

char Capture(int a, int b, int c, int d) {

    return (a == c && b == d) ? 'X' : 0;

}

char pawn(int a, int b, int c, int d) {

    return Capture(a, b, c, d) ? 'X' : (c - a == 1 && abs(d - b) == 1) ? 'P' : 0;

}

char rook(int a, int b, int c, int d) {

    return Capture(a, b, c, d) ? 'X' : (a == c || b == d) ? 'R' : 0;

}

char bishop(int a, int b, int c, int d) {

    return Capture(a, b, c, d) ? 'X' : (abs(a - c) == abs(b - d)) ? 'B' : 0;

}

char knight(int a, int b, int c, int d) {

    return Capture(a, b, c, d) ? 'X' : (abs(a - c) == 2 && abs(d - b) == 1) || (abs(a - c) == 1 && abs(d - b) == 2) ? 'N' : 0;

}

char queen(int a, int b, int c, int d) {

    return Capture(a, b, c, d) ? 'X' : (a == c || b == d || abs(a - c) == abs(b - d)) ? 'Q' : 0;

}

char king(int a, int b, int c, int d) {

    return Capture(a, b, c, d) ? 'X' : (abs(a - c) <= 1 && abs(b - d) <= 1) ? 'K' : 0;

}

bool BoardEmpty(const std::string& str) {

    return std::all\_of(str.begin(), str.end(), [](char c) {

        return c == '.';

    });

}

int main() {

    using namespace std;

    char board[8][8];

    vector<char> sym;

    vector<int> position1;

    vector<int> position2;

    for (int i = 0; i < 8; i++) {

        for (int j = 0; j < 8; j++) {

            cin >> board[i][j];

            if (board[i][j] != 'O') {

                sym.push\_back(board[i][j]);

                position1.push\_back(i + 1);

                position2.push\_back(j + 1);

            }

        }

    }

    int n, x, y;

    cin >> n;

    for (int t = 0; t < n; t++) {

        cin >> x >> y;

        string answer;

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

            char resultChar = 0;

            switch (sym[i]) {

                case 'P':

                    resultChar = pawn(position1[i], position2[i], x, y);

                    break;

                case 'R':

                    resultChar = rook(position1[i], position2[i], x, y);

                    break;

                case 'B':

                    resultChar = bishop(position1[i], position2[i], x, y);

                    break;

                case 'N':

                    resultChar = knight(position1[i], position2[i], x, y);

                    break;

                case 'Q':

                    resultChar = queen(position1[i], position2[i], x, y);

                    break;

                case 'K':

                    resultChar = king(position1[i], position2[i], x, y);

                    break;

            }

            if (resultChar) {

                answer += resultChar;

            }

        }

        if (BoardEmpty(answer)) {

            cout << 'O' << endl;

        } else if (answer.find('X') != string::npos) {

            cout << "X" << endl;

        } else {

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

            auto it = unique(answer.begin(), answer.end());

            answer.erase(it, answer.end());

            cout << answer << endl;

        }

    }

    return 0;

}

Завдання №6 Class Practice Work

#include <iostream>

#include <fstream>

#include <string>

enum File {

    Success,

    Failure,

};

File write\_to\_file(const std::string& name, const std::string& content);

File copy\_file(const std::string& file\_from, const std::string& file\_to);

int main() {

    using namespace std;

    string contest;

    cout << "Enter content for the file: ";

    getline(cin, contest);

    File writeResult = write\_to\_file("from.txt", contest);

    if (writeResult == Success) {

        cout << "File created and written successfully." << endl;

        File copyResult = copy\_file("from.txt", "file\_to.txt");

        if (copyResult == Success) {

            cout << "File contents copy successfully." << endl;

        } else {

            cerr << "File contents copy failed." << endl;

        }

    } else {

        cerr << "File creation or writing failed." << endl;

    }

    return 0;

}

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

    std::ofstream outputFile(name, std::ios::trunc);

    if (outputFile.is\_open()) {

        outputFile << content;

        outputFile.close();

        return Success;

    } else {

        std::cerr << "Could not open file '" << name << "' for writing." << std::endl;

        return Failure;

    }

}

File copy\_file(const std::string& file\_from, const std::string& file\_to) {

    using namespace std;

    std::ifstream from(file\_from);

    if (!from.is\_open()) {

        cerr << "Could not open source file." << endl;

        return Failure;

    }

    std::ofstream to(file\_to);

    using namespace std;

    if (!to.is\_open()) {

        cerr << "Could not open target file." << endl;

        from.close();

        return Failure;

    }

    char c;

    while (from.get(c)) {

        to.put(c);

    }

    from.close();

    to.close();

    return Success;

}

Завдання №7 Self Practice Work

#include <iostream>

#include <set>

#include <string>

#include <unordered\_set>

int main() {

    std::string word;

    std::cin >> word;

    std::unordered\_set<char> letters;

    for (char letter : word) {

        letters.insert(letter);

    }

    std::cout << letters.size() << std::endl;

    return 0;

}

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

Завдання №1 VNS Lab 6

Enter string: 4 Hi 2 my 9 name 1 is 7 Nastia

9

7

4

2

1

Час затрачений на виконання завдання: 60 хвилин

Завдання №2 VNS Lab 8

1984 Orwell 1948 312

The Great Gatsby Fitzgerald 1925 578

War and Peace Leo Tolstoy 1899 416

Pride and Prejudice Jane Austen 1813 238

The Iliad Homer 1598 160

Enter the information about a book you want to add:

Title: Harry Potter

Author: Rowling

Year: 1997

Pages: 560

Before which book do you want to insert new one?

Pride and Prejudice

This book doesn't exist.

Час затрачений на виконання завдання: 3 години

Завдання №3 VNS Lab 9

There are 8 vowels in the first line

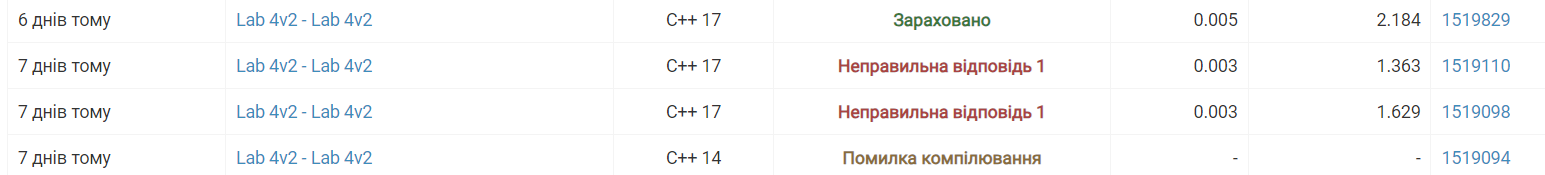
Час затрачений на виконання завдання: 2 години

Завдання №4 Algotester Lab 4

5 2

4 2 7 4 1

4



*Рисунок 2. Результат зарахування програми №4 на Алготестері*

Час затрачений на виконання завдання: 1 година

Завдання №5 Algotester Lab 6

OOOOOOOO

OROOOOOO

OONOOOOO

OOOOPOOO

OOOOOOOO

OOOOOOOO

KOQOOOOO

OOOOOOOR

7

8 1

1 2

NR

5 44

O

5 1

Q

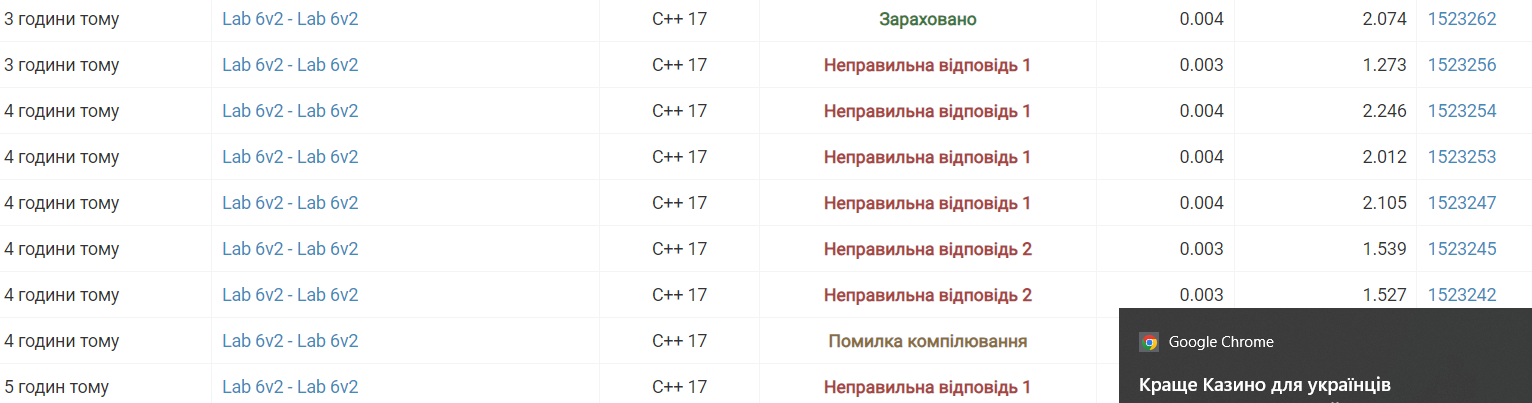
6 2

KQR

8 4

6 7

O



*Рисунок 3. Результат зарахування програми №5 на Алготестері*

Час затрачений на виконання завдання: 5 годин

Завдання №6 Class Practice Work

Enter content for the file: Hi! I'm Nastia

File created and written successfully.

File contents copied successfully.

Завдання №7 Self Practice Work

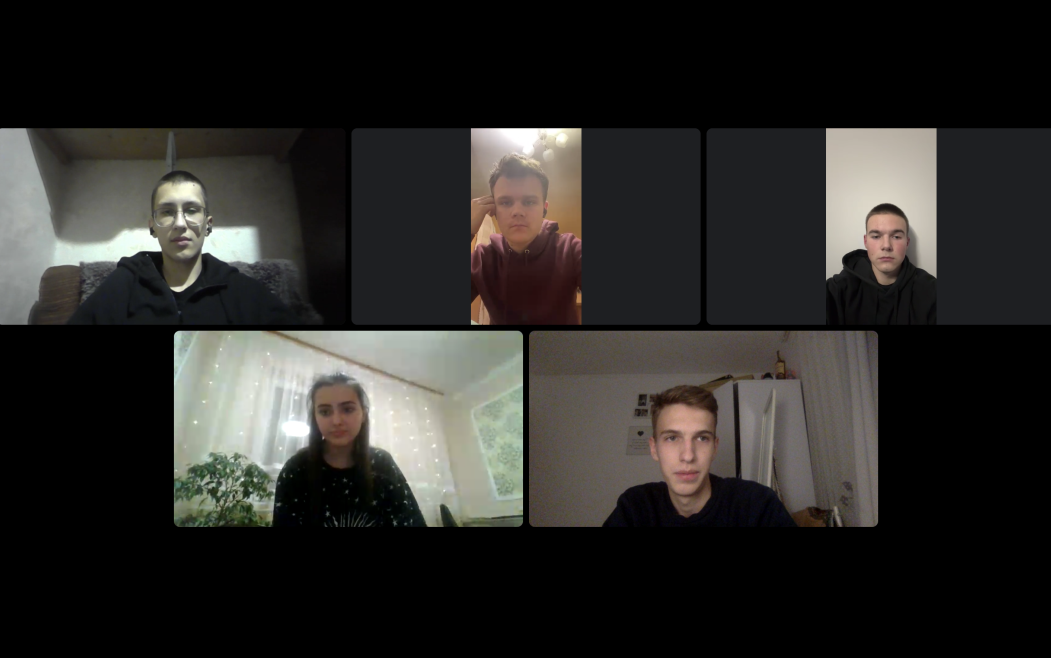
KOLOBOK

4

Час затрачений на виконання завдання: 1 година

## **6. Кооперація з командою:**

* Скрін з зустрічі



*Рисунок 4. Скріншот зустрічі*

# **Висновки:**

У процесі вивчення теоретичного матеріалу та виконання завдань цього розділу я навчилась використовувати файл, рядок і символьні елементи в мові С++. Особливих труднощів при обробці матеріалу не виникало. Через лабораторно-практичні роботи та виконання самостійних занять фактично закріплюються всі пройдені матеріали.