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-12

Виклюк Яна Ярославівна

# **Тема роботи:**

Ознайомлення з файлами, системами числення, бінарними файлами, символами і рядковими змінними та текстовими файлами, стандартною бібліотекою. Розуміння деталей/методів роботи з файлами. Отримання навичок створення й використання бібліотек.

# **Мета роботи:**

Освоїти навички роботи з файлами, системами числення, бінарними файлами, символами і рядковими змінними, а також текстовими файлами за допомогою стандартної бібліотеки. Отримати глибоке розуміння деталей і методів обробки файлів для ефективного використання та створення власних бібліотек.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Файли.
* Тема №2: Бібліотеки.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Файли.
  + Джерела Інформації:
    - <https://acode.com.ua/urok-220-bazovyj-fajlovyj-vvid-i-vyvid/>
    - <http://cpp.dp.ua/ponyattya-fajla/>
    - <https://www.bestprog.net/uk/2019/09/11/examples-of-using-c-tools-for-working-with-files-ua/>
  + Що опрацьовано:
    - Опрацьовано файли, їх види, особливості роботи з ними, потоковий ввід та вивід.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 03.12
* Звершення опрацювання теми: 03.12
* Тема №2 : Бібліотеки.
  + Джерела Інформації
    - <https://acode.com.ua/statychni-i-dynamichni-biblioteky/>
  + Що опрацьовано:
    - Опрацьовано бібліотеки та їх види.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 04.12
  + Звершення опрацювання теми: 04.12

# **Виконання роботи:**

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

Завдання №1 VNS Lab 6

* Варіант №8
* Деталі завдання
* Перетворити рядок так, щоб всі слова в ньому стали ідентифікаторами, слова
* які складаються тільки із цифр - знищити.
* Важливі деталі для врахування в імплементації програми
* Слово є ідентифікатором, якщо воно не починається з цифри, тобто якщо на початку слова є цифра його необхідно видалити.

Завдання №2 VNS Lab 8

* Варіант №8
* Деталі завдання
* Сформувати двійковий файл із елементів, структури "Покупець",
* 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.
* Важливі деталі для врахування в імплементації програми
* Врахувати особливості роботи з файлами, передбачити можливі помилки.

Завдання №3 VNS Lab 9

* Варіант №8
* Деталі завдання
* Скопіювати з файлу F1 у файл F2 всі рядки, які не містять цифри.
* Підрахувати кількість рядків, які починаються на букву «А» у файлі F2.
* Важливі деталі для врахування в імплементації програми
* Зрозуміти, що потрібно пройтися по кожному символу рядка та перевірити чи є він числом, перевірити лише перший символ рядка для другого завдання.

Завдання №4 Algotester Lab 4.1

* Варіант №1
* Умова
* Вам дано 2 цілих чисел масиви, розміром N та M.
* Ваше завдання вивести:
* 1. Різницю N-M
* 2. Різницю M-N
* 3. Їх перетин
* 4. Їх обєднання
* 5. Їх симетричну різницю

# Вхідні дані

* У першому рядку ціле число N - розмір масиву 1
* У другому рядку N цілих чисел - елементи масиву 1
* У третьому рядку ціле число M - розмір масиву 2
* У четвертом рядку M цілих чисел - елементи масиву 2
* Деталі завдання
* Вивести результат виконання 5 вищезазначених операцій у форматі:
* У першому рядку ціле число N - розмір множини
* У наступному рядку N цілих чисел - посортована у порядку зростання множина
* Важливі деталі для врахування в імплементації програми
* Врахувати, що важливо відсортувати масиви до того як робити операції над ними.

Завдання №5 Algotester Lab 4.2

* Варіант №1
* Умова
* Вам дано 2 цілих чисел масиви, розміром N та M.
* Ваше завдання вивести:
* 1. Різницю N-M
* 2. Різницю M-N
* 3. Їх перетин
* 4. Їх обєднання
* 5. Їх симетричну різницю
* Без використання засобів STL

# Вхідні дані

* У першому рядку ціле число N - розмір масиву 1
* У другому рядку N цілих чисел - елементи масиву 1
* У третьому рядку ціле число M - розмір масиву 2
* У четвертом рядку M цілих чисел - елементи масиву 2
* Деталі завдання
* Вивести результат виконання 5 вищезазначених операцій у форматі:
* У першому рядку ціле число N - розмір множини
* У наступному рядку N цілих чисел - посортована у порядку зростання множина
* Важливі деталі для врахування в імплементації програми
* Врахувати, що важливо відсортувати масиви до того як робити операції над ними, об’єднання це відсортовані дві різниці та перетин, а симетрична різниця це відсортовані дві різниці.

Завдання №6 Algotester Lab 6

* Варіант №1
* Умова
* Вам дано N слів та число K.
* Ваше завдання перечислити букви в словах, які зустрічаються в тексті більше-рівне ніж K разів (саме слово, не буква!).
* Великі та маленькі букви вважаються однаковими, виводити необхідно малі, посортовані від останьої до першої у алфавіті. Букву потрібно виводити лише один раз.
* У випадку якщо таких букв немає - вивести "Empty!".
* Вхідні дані
* Цілі числа N та K - загальна кількість слів та мінімальна кількість слів щоб враховувати букви цього слова в результаті.
* N стрічок s
* Деталі завдання
* Вивести у першому рядку ціле число M - кількість унікальних букв, у другому рядку унікальні букви через пробіли.
* Важливі деталі для врахування в імплементації програми
* Врахувати, що для оптимізованої програми потрібно застосовувати unordered\_map.

Завдання №7 Class Practice Work

* Варіант завдання відсутній
* Деталі завдання
* Реалізувати функцію створення файла і запису в нього даних, функцію копіювання вмісту файла у інший файл.
* Важливі деталі для врахування в імплементації програми
* Врахувати особливості роботи з файлами.

Завдання №8 Self Practice Work (Algotester lab2v3)

* Варіант завдання відсутній
* Умова
* Вам дано масив цілих чисел розміром N, на першій та останній клітинці розміщено по дрону.
* Вони одночасно взлітають.
* На початку кожного ходу швидкість дрону стає рівною значенню клітинки, у якій він знаходиться.
* Тобто лівий дрон у першу секунду з клітинки з індексом 1 перелетить у клітинку з індексом a1, тобто його наступна позиція рахується як поточна позиція + число у поточній позиції (перегляньте пояснення для візуалізації) Правий робить аналогічно в протилежну сторону.
* Вони це роблять до моменту, коли трапиться одна з зазначених подій:
* Якшо 2 дрони опиняються в одній клітинці - ви виводите **Collision**.
* Якщо лівий дрон опиниться справа від правого - це **Miss**
* У випадку якщо вони зупиняться один навпроти одного, тобто у клітинках ai та ai+1 - виведіть **Stopped**
* Врахуйте, що перевіряти треба також до взльоту.
* Вхідні дані
* У першому рядку ціле число N - розмір масиву
* У другому рядку N цілих чисел - елементи масиву
* Деталі завдання
* Вивести у першому рядку фінальна позиція першого та другого дрона.
* У другому рядку одне зі слів:
* **Collision**
* **Miss**
* **Stopped**
* Важливі деталі для врахування в імплементації програми
* Обережно працювати з масивом, щоб не вилізти за нього, з обох боків.

Завдання №9 Self Practice Work (Algotester lab3v2)

* Варіант 1
* Умова
* Вам дано 2 масиви розміром N та M. Значення у цих масивах унікальні.
* Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.
* Вхідні дані
* У першому рядку ціле число N
* у другому рядку N цілих чисел a1..an
* У третьому рядку ціле число M
* у четвертому рядку M цілих чисел b1..bn
* Деталі завдання
* Вивести у першому рялку одне ціле число - кількість елементів, які наявні в обох масивах одночасно, у другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об’єднанням двох даних).
* Важливі деталі для врахування в імплементації програми
* Врахувати, що кількість унікальних елементів буде дорівнювати сумі довжин двох масивів відняти кількість елементів, котрі є у обох масивах водночас.

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

Програма №1 VNS Lab 6

* Блок-схема

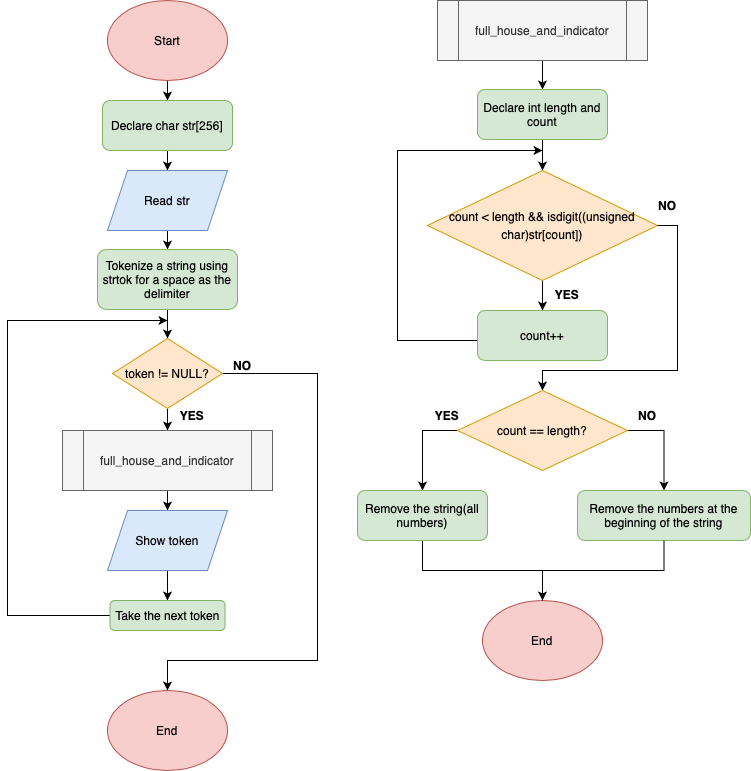


Рисунок 1 Блок-схема до програми №1

* Планований час на реалізацію 60 хв
* Важливі деталі для врахування в імплементації
* Слово є ідентифікатором, якщо воно не починається з цифри, тобто якщо на початку слова є цифра його необхідно видалити.

Програма №2 VNS Lab 8

* Блок-схема
* Планований час на реалізацію 2 год
* Важливі деталі для врахування в імплементації
* Врахувати особливості роботи з файлами, передбачити можливі помилки.

Програма №3 VNS Lab 9

* Планований час на реалізацію 2 год
* Важливі деталі для врахування в імплементації
* Зрозуміти, що потрібно пройтися по кожному символу рядка та перевірити чи є він числом, перевірити лише перший символ рядка для другого завдання.

Програма №4 Algotester Lab 4.1

* Планований час на реалізацію 90 хв
* Важливі деталі для врахування в імплементації
* Врахувати, що важливо відсортувати масиви до того як робити операції над ними.

Програма №5 Algotester Lab 4.2

* Планований час на реалізацію 90 хв
* Важливі деталі для врахування в імплементації
* Врахувати, що важливо відсортувати масиви до того як робити операції над ними, об’єднання це відсортовані дві різниці та перетин, а симетрична різниця це відсортовані дві різниці.

Програма №6 Algotester Lab 6

* Планований час на реалізацію 2 год
* Важливі деталі для врахування в імплементації
* Врахувати, що для оптимізованої програми потрібно застосовувати unordered\_map.

Програма №7 Class Practice Work

* Планований час на реалізацію 45 хв
* Важливі деталі для врахування в імплементації
* Врахувати особливості роботи з файлами.

Програма №8 Self Practice Work (Algotester lab2v3)

* Планований час на реалізацію 30 хв
* Важливі деталі для врахування в імплементації
* Обережно працювати з масивом, щоб не вилізти за нього, з обох боків.

Програма №9 Self Practice Work (Algotester lab3v2)

* Планований час на реалізацію 40 хв
* Важливі деталі для врахування в імплементації
* Врахувати, що кількість унікальних елементів буде дорівнювати сумі довжин двох масивів відняти кількість елементів, котрі є у обох масивах водночас.

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

Деталі по конфігурації середовища

Достатньо базових розширень та налаштувань Visual Studio Code, C++, нічого додаткового не потрібно для кожного завдання.

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

Завдання №1 VNS Lab 6

Деталі по програмі

Перетворити рядок так, щоб всі слова в ньому стали ідентифікаторами, слова які складаються тільки із цифр – знищити.

#include <iostream>

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

#include <cstdio>

using namespace std;

void full\_house\_and\_indicator(char \*str)

{

int length = strlen(str);

int count = 0;

while (count < length && isdigit((unsigned char)str[count]))

{

count++;

}

if(count == length)

{

str[0] = '\0';

}

else

{

memmove(str, str + count, length - count + 1);

}

};

int main()

{

char str[256];

fgets(str, 255, stdin);

char\*token;

token = strtok(str, " ");

while(token != NULL)

{

full\_house\_and\_indicator(token);

cout << token << " ";

token = strtok(NULL, " ");

}

cout << endl;

return 0;

}

Блок з кодом до завдання №1

Завдання №2 VNS Lab 8

Деталі по програмі

Сформувати двійковий файл із елементів, структури "Покупець", 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.

#include <iostream>

#include <cstdio>

using namespace std;

#define n 4

struct Buyer

{

char name[100];

char adress[100];

int number;

int credit\_card;

};

void form()

{

FILE \*f;

if ((f = fopen("f.txt", "wb")) == NULL)

{

perror("Error opening file for writing");

exit(1);

}

Buyer e[n];

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

{

// form

printf("Enter name = ");

scanf("%s", e[i].name);

printf("Enter adress = ");

scanf("%s", e[i].adress);

printf("Enter phone number = ");

scanf("%d", &e[i].number);

printf("Enter credit card = ");

scanf("%d", &e[i].credit\_card);

// write in file

fwrite(&e[i], sizeof(Buyer), 1, f);

}

fclose(f);

}

void print()

{

FILE \*f = fopen("f.txt", "rb");

if (f == NULL)

{

perror("Error opening file for reading");

exit(1);

}

Buyer e[n];

fread(e, sizeof(Buyer), n, f);

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

{

printf("name = %s \n", e[i].name);

printf("adress = %s \n", e[i].adress);

printf("phone number = %d \n", e[i].number);

printf("credit card = %d \n", e[i].credit\_card);

}

fclose(f);

}

void delete\_elements()

{

FILE \*f = fopen("f.txt", "rb");

if (f == NULL)

{

perror("Error opening file for reading");

exit(1);

}

Buyer e[n];

int total;

fseek(f, 0, SEEK\_END);

total = ftell(f) / sizeof(Buyer);

rewind(f);

if (total <= 3)

{

fclose(f);

f = fopen("f.txt", "wb");

fclose(f);

return;

}

// read from file to array

fread(e, sizeof(Buyer), n, f);

fclose(f);

// erase

for (int i = 0; i < total - 3; i++)

{

e[i] = e[i + 3];

}

f = fopen("f.txt", "wb");

if (!f)

{

perror("Error opening file for writing");

exit(1);

}

fwrite(e, sizeof(Buyer), total - 3, f);

fclose(f);

}

void add\_elements()

{

FILE \*f = fopen("f.txt", "ab");

if (f == NULL)

{

perror("Error opening file for appending");

exit(1);

}

Buyer new\_buyers[3];

for (int i = 0; i < 3; i++)

{

cout << "Enter new name: ";

scanf("%s", new\_buyers[i].name);

cout << "Enter new address: ";

scanf("%s", new\_buyers[i].adress);

cout << "Enter new phone number: ";

scanf("%d", &new\_buyers[i].number);

cout << "Enter new credit card: ";

scanf("%d", &new\_buyers[i].credit\_card);

// write the new buyer to the end of the file

fwrite(&new\_buyers[i], sizeof(Buyer), 1, f);

}

fclose(f);

}

int main()

{

form();

delete\_elements();

add\_elements();

cout << "New : \n" << endl;

print();

return 0;

}

Блок з кодом до завдання №2

Завдання №3 VNS Lab 9

Деталі по програмі

Скопіювати з файлу F1 у файл F2 всі рядки, які не містять цифри. Підрахувати кількість рядків, які починаються на букву «А» у файлі F2.

#include <iostream>

#include <fstream>

#include <string>

#include <cctype>

using namespace std;

void create\_first\_file()

{

ofstream outf("F1.txt");

if(!outf)

{

cerr << "Error" << endl;

exit(1);

}

outf << "Last Christmas I gave you my heart\n";

outf << "But the very next day you gave it away\n";

outf << "T1his year, to sa23ve 12me fr1om tears\n";

outf << "213I'll giv13e it 3to someone sp42ecial, special\n";

outf << "La34st Christ12mas I ga1ve 1you my h2eart\n";

outf << "But 3948 the 24 very 24 next75 day you gav14e it away\n";

outf << "This year, to save me from tears\n";

outf << "I'll give it to someone special\n";

outf << "Spe76c8ial\n";

outf << "A face on a lover with a fire in his heart (I gave you mine)\n";

outf << "A man under cover but you tore him apart\n";

outf << "Maybe next year I'll give it to someone\n";

outf << "I'll give it to someone special\n";

outf.close();

};

void copy\_in\_file()

{

ifstream inf("F1.txt");

ofstream outf("F2.txt");

if(!outf || !inf)

{

cerr << "Error" << endl;

exit(1);

}

string line;

while(getline(inf, line))

{

bool number = false;

for(string:: iterator it = line.begin(); it != line.end(); ++it)

{

if(isdigit(\*it))

{

number = true;

}

}

if(!number)

{

outf << line << endl;

}

}

inf.close();

outf.close();

};

void copy\_in\_file\_add()

{

ifstream inf("F1.txt");

ofstream outf("F2.txt");

if(!outf || !inf)

{

cerr << "Error" << endl;

exit(1);

}

string line;

while(getline(inf, line))

{

for(string:: iterator it = line.begin(); it != line.end(); ++it)

{

if(isdigit(\*it))

{

it = line.erase(it);

it--;

}

}

outf << line << endl;

}

inf.close();

outf.close();

};

void counter()

{

ifstream inf("F2.txt");

if(!inf)

{

cerr << "Error" << endl;

exit(1);

}

string line;

int count = 0;

while(getline(inf, line))

{

string:: iterator it = line.begin();

if(\*it == 'A')

{

count++;

}

}

inf.close();

cout << "The number of lines that start with the letter 'A': " << count << endl;

};

int main()

{

int choose;

create\_first\_file();

cout << "Do you want to copy all lines that do not contain numbers - 1 or to delete all numbers in lines - 2?" << endl;

cin >> choose;

if(choose == 1)

{

copy\_in\_file();

}

if(choose == 2)

{

copy\_in\_file\_add();

}

counter();

return 0;

}

Блок з кодом до завдання №3

Завдання №4 Algotester Lab 4.1

Деталі по програмі

Вивести результат виконання 5 операцій(різниця N-M, різниця M-N, їх перетин, їх обєднання, їх симетричну різниця) у форматі:

У першому рядку ціле число N - розмір множини

У наступному рядку N цілих чисел - посортована у порядку зростання множина.

#include <iostream>

#include <set>

#include <algorithm>

#include <vector>

using namespace std;

int main()

{

int N, M, number\_n, number\_m;

vector <int> n;

vector <int> m;

cin >> N;

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

{

cin >> number\_n;

n.push\_back(number\_n);

}

cin >> M;

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

{

cin >> number\_m;

m.push\_back(number\_m);

}

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

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

vector <int> difference1;

set\_difference(n.begin(), n.end(), m.begin(), m.end(), inserter(difference1, difference1.begin()));

cout << difference1.size() << endl;

for(int i = 0; i < difference1.size(); i++)

{

cout << difference1[i] << " ";

}

cout << endl << "\n";

vector <int> difference2;

set\_difference(m.begin(), m.end(), n.begin(), n.end(), inserter(difference2, difference2.begin()));

cout << difference2.size() << endl;

for(int i = 0; i < difference2.size(); i++)

{

cout << difference2[i] << " ";

}

cout << endl << "\n";

vector <int> crossing;

set\_intersection(n.begin(), n.end(), m.begin(), m.end(), inserter(crossing, crossing.begin()));

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

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

{

cout << crossing[i] << " ";

}

cout << endl << "\n";

vector <int> unification;

set\_union(n.begin(), n.end(), m.begin(), m.end(), inserter(unification, unification.begin()));

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

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

{

cout << unification[i] << " ";

}

cout << endl << "\n";

vector <int> symmetric;

set\_symmetric\_difference(n.begin(), n.end(), m.begin(), m.end(), inserter(symmetric, symmetric.begin()));

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

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

{

cout << symmetric[i] << " ";

}

cout << endl << "\n";

return 0;

}

Блок з кодом до завдання №4

Завдання №5 Algotester Lab 4.2

Деталі по програмі

Вивести результат виконання 5 операцій(різниця N-M, різниця M-N, їх перетин, їх обєднання, їх симетричну різниця) у форматі:

У першому рядку ціле число N - розмір множини

У наступному рядку N цілих чисел - посортована у порядку зростання множина.

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

int main()

{

int N, M, number\_n, number\_m;

vector <int> n;

vector <int> m;

cin >> N;

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

{

cin >> number\_n;

n.push\_back(number\_n);

}

cin >> M;

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

{

cin >> number\_m;

m.push\_back(number\_m);

}

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

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

vector <int> difference1;

vector <int> m\_copy = m;

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

{

bool is = true;

for(int j = 0; j < m\_copy.size(); j++)

{

if(n[i] == m\_copy[j])

{

m\_copy[j] = -1;

is = false;

break;

}

}

if(is)

{

difference1.push\_back(n[i]);

}

}

cout << difference1.size() << endl;

for(int i = 0; i < difference1.size(); i++)

{

cout << difference1[i] << " ";

}

cout << endl << "\n";

vector <int> difference2;

vector <int> n\_copy = n;

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

{

bool is = true;

for(int j = 0; j < n\_copy.size(); j++)

{

if(m[i] == n\_copy[j])

{

n\_copy[j] = -1;

is = false;

break;

}

}

if(is)

{

difference2.push\_back(m[i]);

}

}

cout << difference2.size() << endl;

for(int i = 0; i < difference2.size(); i++)

{

cout << difference2[i] << " ";

}

cout << endl << "\n";

vector <int> crossing;

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

{

if(n\_copy[i] == -1)

{

crossing.push\_back(n[i]);

}

}

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

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

{

cout << crossing[i] << " ";

}

cout << endl << "\n";

vector <int> unification;

for(int i = 0; i < difference1.size(); i++)

{

unification.push\_back(difference1[i]);

}

for(int i = 0; i < difference2.size(); i++)

{

unification.push\_back(difference2[i]);

}

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

{

unification.push\_back(crossing[i]);

}

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

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

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

{

cout << unification[i] << " ";

}

cout << endl << "\n";

vector <int> symmetric;

for(int i = 0; i < difference1.size(); i++)

{

symmetric.push\_back(difference1[i]);

}

for(int i = 0; i < difference2.size(); i++)

{

symmetric.push\_back(difference2[i]);

}

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

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

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

{

cout << symmetric[i] << " ";

}

cout << endl << "\n";

return 0;

}

Блок з кодом до завдання №5

Завдання №6 Algotester Lab 6

Деталі по програмі

Вивести у першому рядку ціле число M - кількість унікальних букв, у другому рядку унікальні букви через пробіли.

#include <iostream>

#include <string>

#include <unordered\_map>

#include <set>

#include <algorithm>

#include <vector>

using namespace std;

int main()

{

int N, K;

cin >> N >> K;

vector<string> s;

set <char> letters;

unordered\_map<string, int> count;

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

{

string si;

cin >> si;

if (si.size() < 1 || si.size() > 10)

{

return 0;

}

transform(si.begin(), si.end(), si.begin(), ::tolower);

s.push\_back(si);

count[si]++;

}

for (unordered\_map<string, int> :: iterator it = count.begin(); it != count.end(); ++it)

{

if (it -> second >= K)

{

const string& word = it -> first;

for (string::const\_iterator charIt = word.begin(); charIt != word.end(); ++charIt)

{

letters.insert(\*charIt);

}

}

}

if (letters.empty())

{

cout << "Empty!\n";

return 0;

}

cout << letters.size() << "\n";

for (set <char> :: reverse\_iterator it = letters.rbegin(); it != letters.rend(); ++it)

{

cout << \*it << ' ';

}

cout << '\n';

return 0;

}

Блок з кодом до завдання №6

Завдання №7 Class Practice Work

Деталі по програмі

Реалізувати функцію створення файла і запису в нього даних, функцію копіювання вмісту файла у інший файл.

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

enum FileOpResult

{

Success,

Failure

};

FileOpResult write\_to\_file(string name, string content)

{

ofstream outf(name);

if(!outf)

{

cerr << "Error" << endl;

return Failure;

}

outf << content << endl;

outf.close();

return Success;

};

FileOpResult copy\_file(string file\_from, string file\_to)

{

ifstream inf(file\_from);

ofstream outf(file\_to);

if(!outf || !inf)

{

cerr << "Error" << endl;

return Failure;

}

string line;

while(getline(inf, line))

{

outf << line << endl;

}

inf.close();

outf.close();

return Success;

};

int main()

{

cout << "Write the name of the file, you want to create" << endl;

string name, content;

cin >> name;

cout << "What do you want to add there?" << endl;

cin.ignore();

getline(cin, content);

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

if(result\_write == Success)

{

cout << "First task is successfully done" << endl;

}

else

{

cout << "It was a fatal failure" << endl;

}

cout << "Write the name of the file, you want to copy from" << endl;

string file\_from, file\_to;

cin >> file\_from;

cout << "Write the name of the file, you want to copy to" << endl;

cin >> file\_to;

FileOpResult result\_copy = copy\_file(file\_from, file\_to);

if(result\_copy == Success)

{

cout << "Second task is successfully done" << endl;

}

else

{

cout << "It was a fatal failure 2x" << endl;

}

return 0;

}

Блок з кодом до завдання №7

Завдання №8 Self Practice Work (Algotester lab2v3)

Деталі по програмі

Вивести у першому рядку фінальна позиція першого та другого дрона.

У другому рядку одне зі слів:

* **Collision**
* **Miss**
* **Stopped**

#include <iostream>

using namespace std;

int main()

{

int N;

cin >> N;

const int n = N;

int a[n];

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

{

cin >> a[i];

}

int left\_dron = 0;

int right\_dron = n - 1;

while(left\_dron < right\_dron)

{

int left = a[left\_dron];

int right = a[right\_dron];

if(left\_dron == right\_dron - 1)

{

cout << left\_dron + 1 << " " << right\_dron + 1 << endl;

cout << "Stopped" << endl;

break;

}

left\_dron += left;

right\_dron -= right;

}

if(left\_dron == right\_dron)

{

cout << left\_dron + 1 << " " << right\_dron + 1 << endl;

cout << "Collision" << endl;

}

if(left\_dron > right\_dron)

{

cout << left\_dron + 1 << " " << right\_dron + 1 << endl;

cout << "Miss" << endl;

}

return 0;

}

Блок з кодом до завдання №8

Завдання №9 Self Practice Work (Algotester lab3v2)

Деталі по програмі

Вивести у першому рялку одне ціле число - кількість елементів, які наявні в обох масивах одночасно, у другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об’єднанням двох даних).

#include <iostream>

using namespace std;

int main()

{

int N, M;

int count = 0;

int unique;

cin >> N;

int a[N];

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

{

cin >> a[i];

}

cin >> M;

int b[M];

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

{

cin >> b[i];

}

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

{

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

{

if(a[i] == b[j])

{

++count;

}

}

}

unique = N + M - count;

cout << count << endl;

cout << unique << endl;

return 0;

}

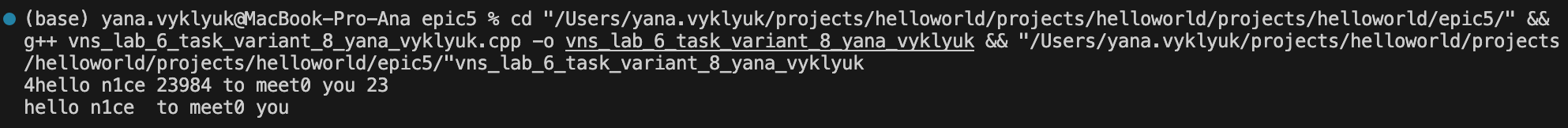
Блок з кодом до завдання №9

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

Завдання №1 VNS Lab 6

Деталі по виконанню і тестуванню програми

При виконанні цієї програми користувач вводить рядок, програма перетворює всі слова на ідентифікатори, тобто видаляє цифри якщо вони стоять на початку слова, а також видаляє слова, які складаються тільки з цифр.



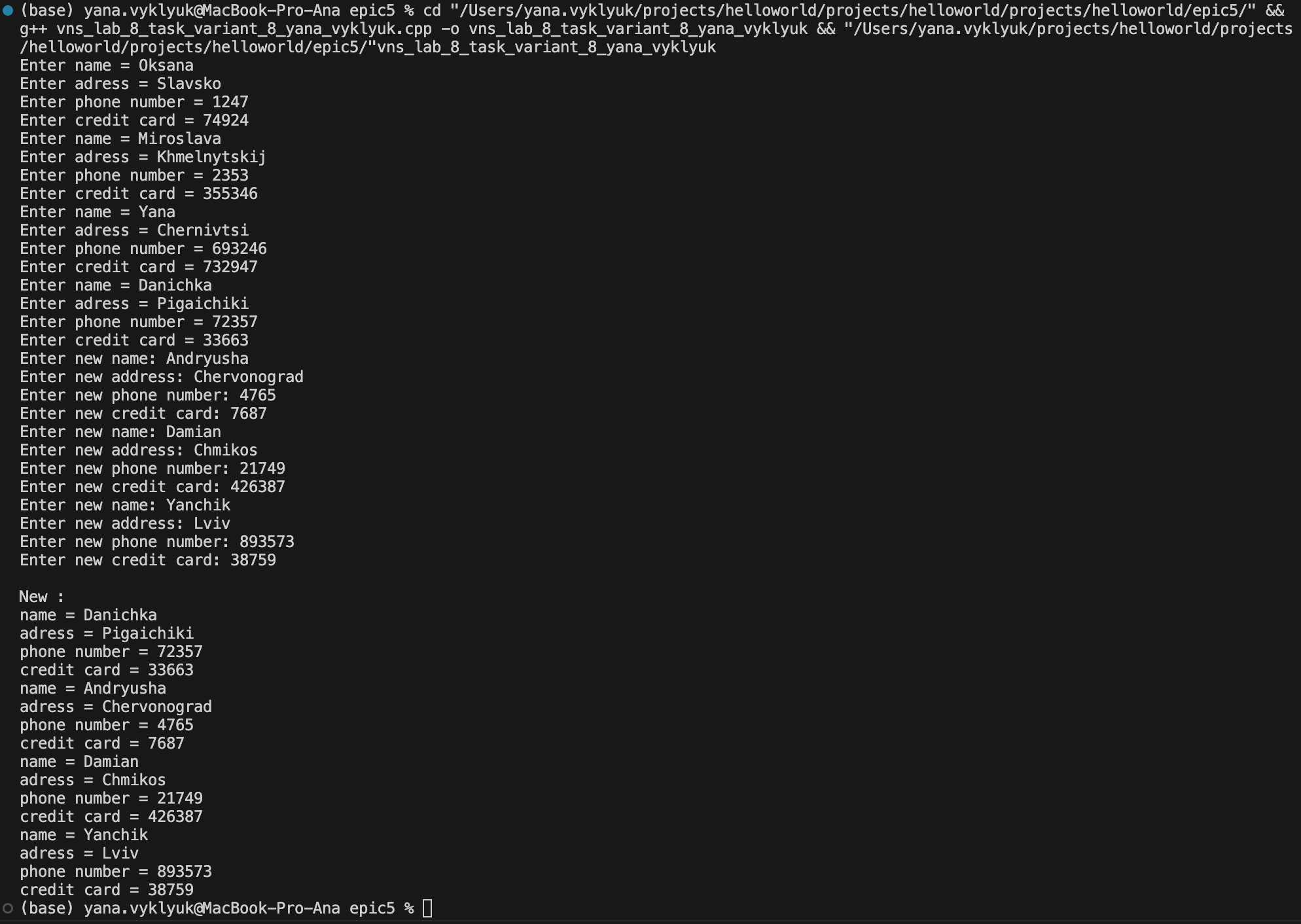
Блок з виконанням та тестуванням програми до завдання №1

Час затрачений на виконання завдання 60 хв.

Завдання №2 VNS Lab 8

Деталі по виконанню і тестуванню програми

Програма формує двійковий файл із елементів, структури "Покупець", котрі вводить користувач, знищує 3 елементи з початку файлу, додає 3 елементи в кінець файлу, використовуючи функцію та виводить з того файлу нові відредаговані елементи. У разі помилки при відкритті файлу й виконанні операцій вводу/виводу програма виводить повідомлення про це.



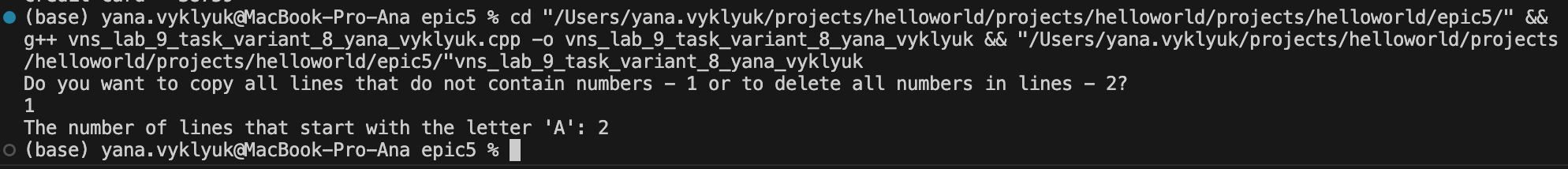
Блок з виконанням та тестуванням програми до завдання №2

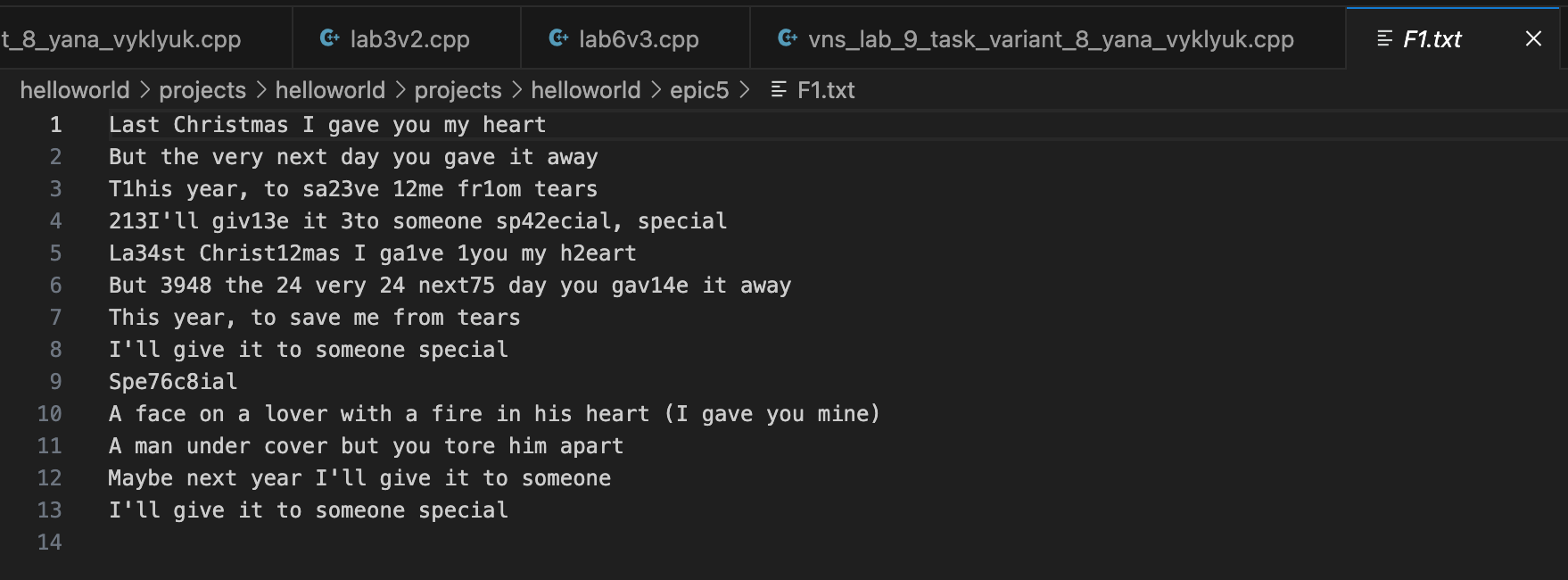
Час затрачений на виконання завдання 2 год

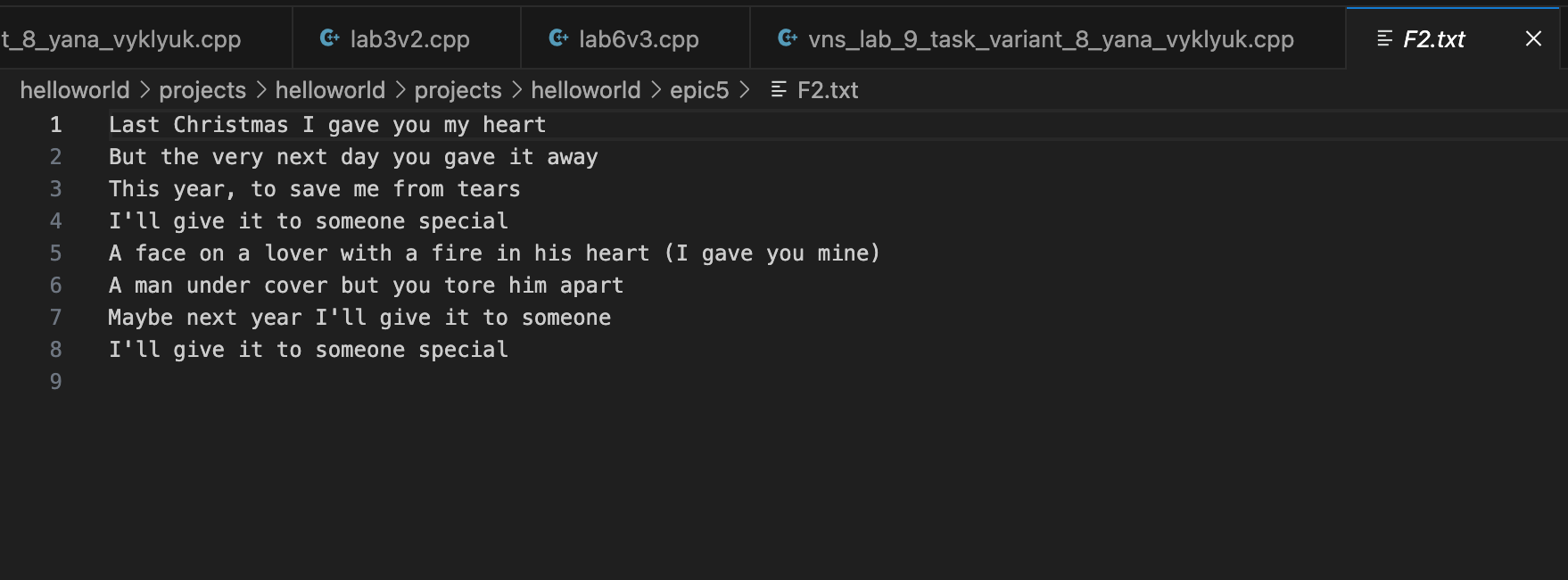
Завдання №3 VNS Lab 9

Деталі по виконанню і тестуванню програми

Програма копіює з файлу F1 у файл F2 всі рядки, які не містять цифри, рахує кількість рядків, які починаються на букву «А» у файлі F2. Також, зроблена додаткова функція, котра видаляє всі числа зі стрічки.







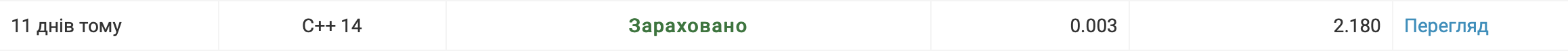
Блок з виконанням та тестуванням програми до завдання №3

Час затрачений на виконання завдання 90 хв

Завдання №4 Algotester Lab 4.1

Деталі по виконанню і тестуванню програми

Код програми загружено на Algotester. Програма виводить результат виконання 5 операцій (різниця N-M, різниця M-N, їх перетин, їх обєднання, їх симетричну різниця), використовуючи засоби STL.





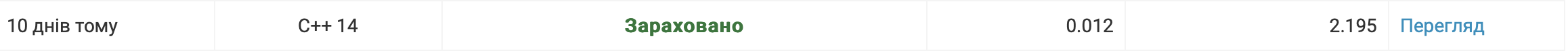
Блок з виконанням та тестуванням програми до завдання №4

Час затрачений на виконання завдання 90 хв.

Завдання №5 Algotester Lab 4.2

Деталі по виконанню і тестуванню програми

Код програми загружено на Algotester. Програма виводить результат виконання 5 операцій (різниця N-M, різниця M-N, їх перетин, їх обєднання, їх симетричну різниця), не використовуючи засоби STL, а використовуючи власну реалізацію.





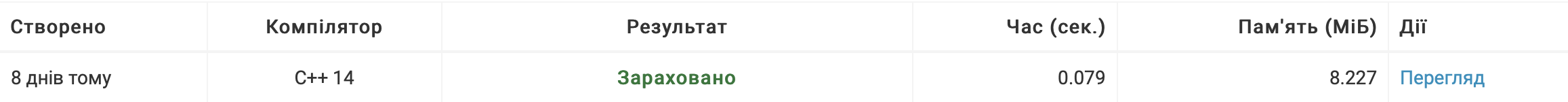
Блок з виконанням та тестуванням програми до завдання №5

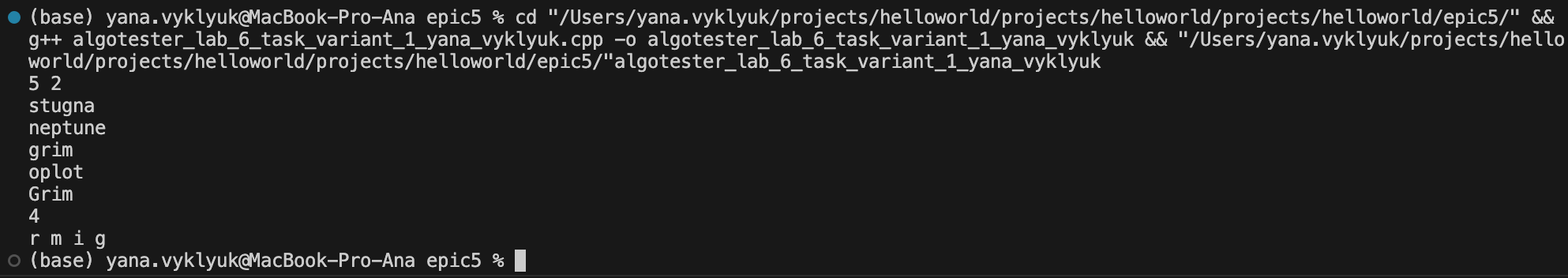
Час затрачений на виконання завдання 90 хв.

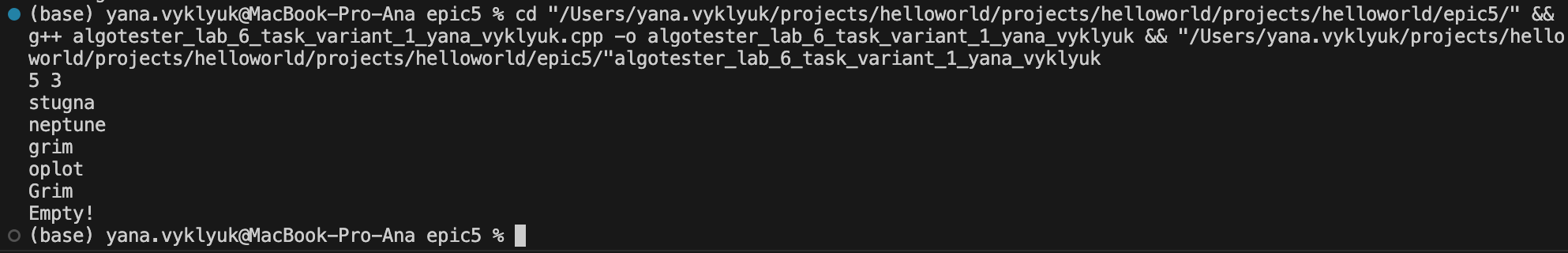
Завдання №6 Algotester Lab 6

Деталі по виконанню і тестуванню програми

Код програми загружено на Algotester. Програма виводить у першому рядку ціле число M - кількість унікальних букв, у другому рядку унікальні букви через пробіли. Для цього спочатку за допомогою unordered\_map підраховує кількість кожного слова, обирає ті які зустрічаються достатню кількість разів, робить з їх букв set, бо мають бути лише унікальні букви, та сортує як вказано в умові. Також, враховується, що малі і великі літері однакові, це робиться при введені слів, тоді усі великі літери записуються як малі.







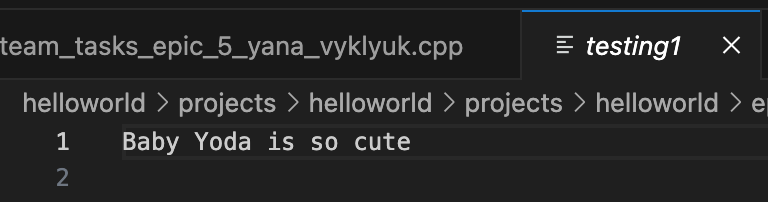
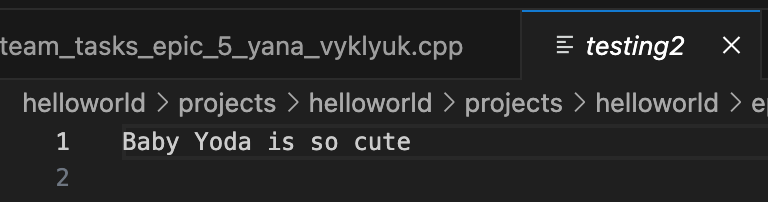
Блок з виконанням та тестуванням програми до завдання №6

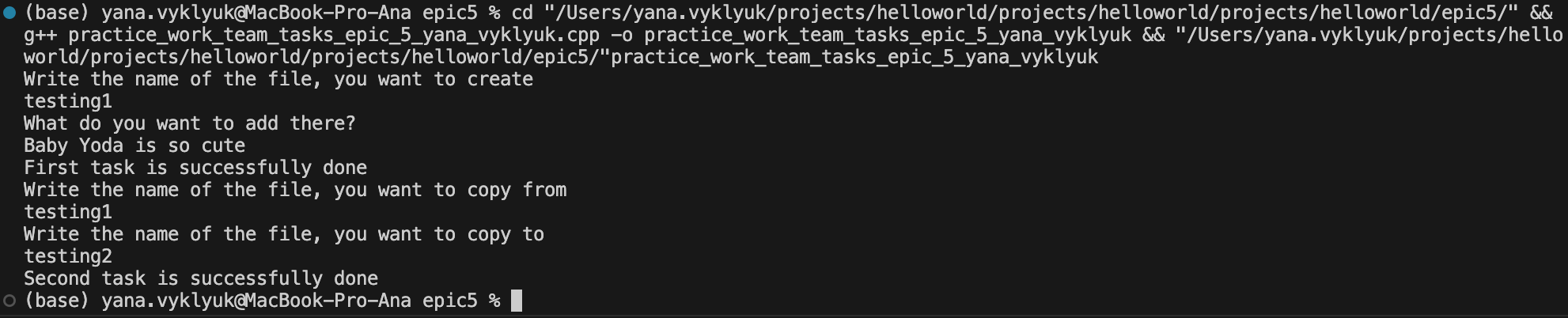
Час затрачений на виконання завдання 2 год.

Завдання №7 Class Practice Work

Деталі по виконанню і тестуванню програми

Програма створює файл і записує в нього дані, копіює вміст файла у інший файл.



Блок з виконанням та тестуванням програми до завдання №7

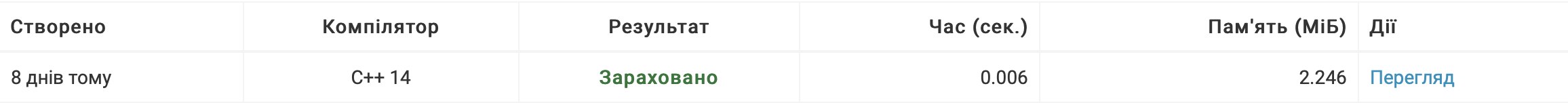
Час затрачений на виконання завдання 45 хв.

Завдання №8 Self Practice Work (Algotester lab2v3)

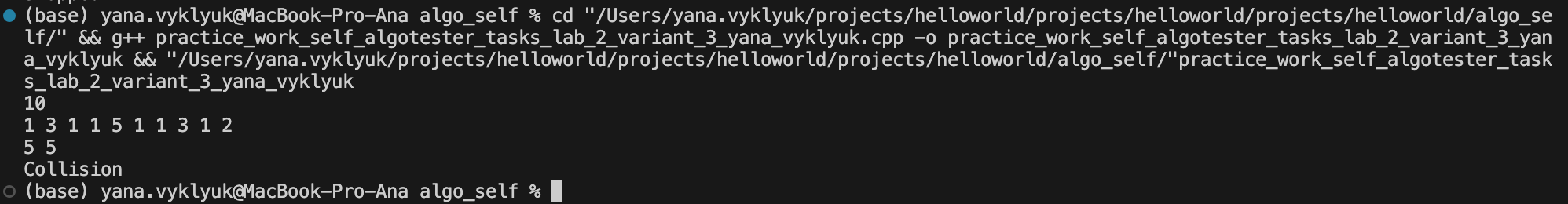
Деталі по виконанню і тестуванню програми

Код програми загружено на Algotester. Програма виводить у першому рядку фінальну позицію першого та другого дрона.

* У другому рядку одне зі слів:
* **Collision**
* **Miss**
* **Stopped**







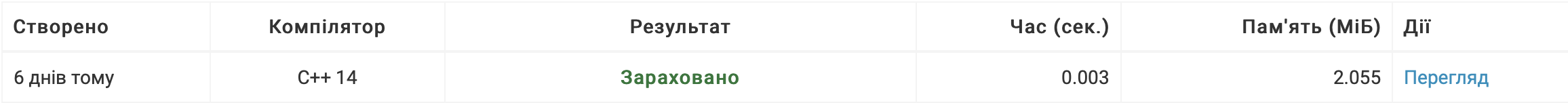
Блок з виконанням та тестуванням програми до завдання №8

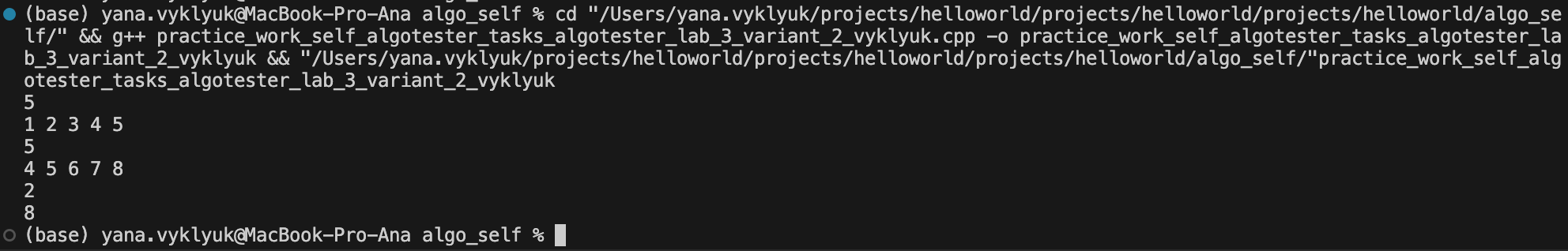
Час затрачений на виконання завдання 30 хв.

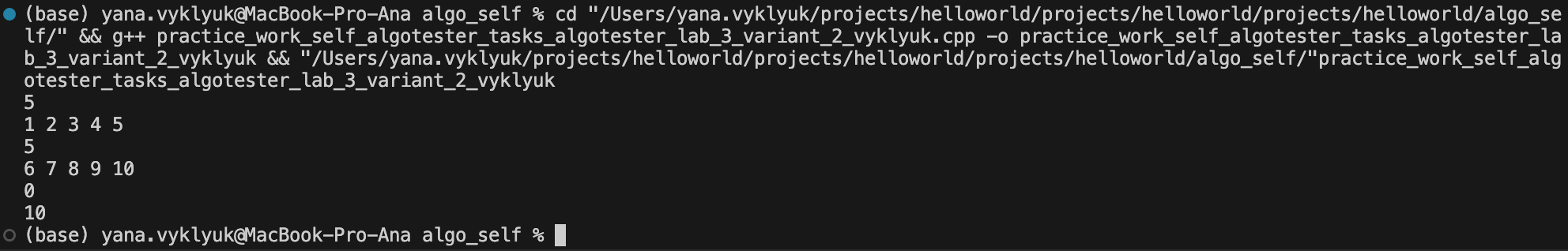
Завдання №9 Self Practice Work (Algotester lab3v2)

Деталі по виконанню і тестуванню програми

Код програми загружено на Algotester. Код програми виводить у першому рялку одне ціле число - кількість елементів, які наявні в обох масивах одночасно, у другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об’єднанням двох даних). Враховано, що кількість унікальних елементів буде дорівнювати сумі довжин двох масивів відняти кількість елементів, котрі є у обох масивах водночас.





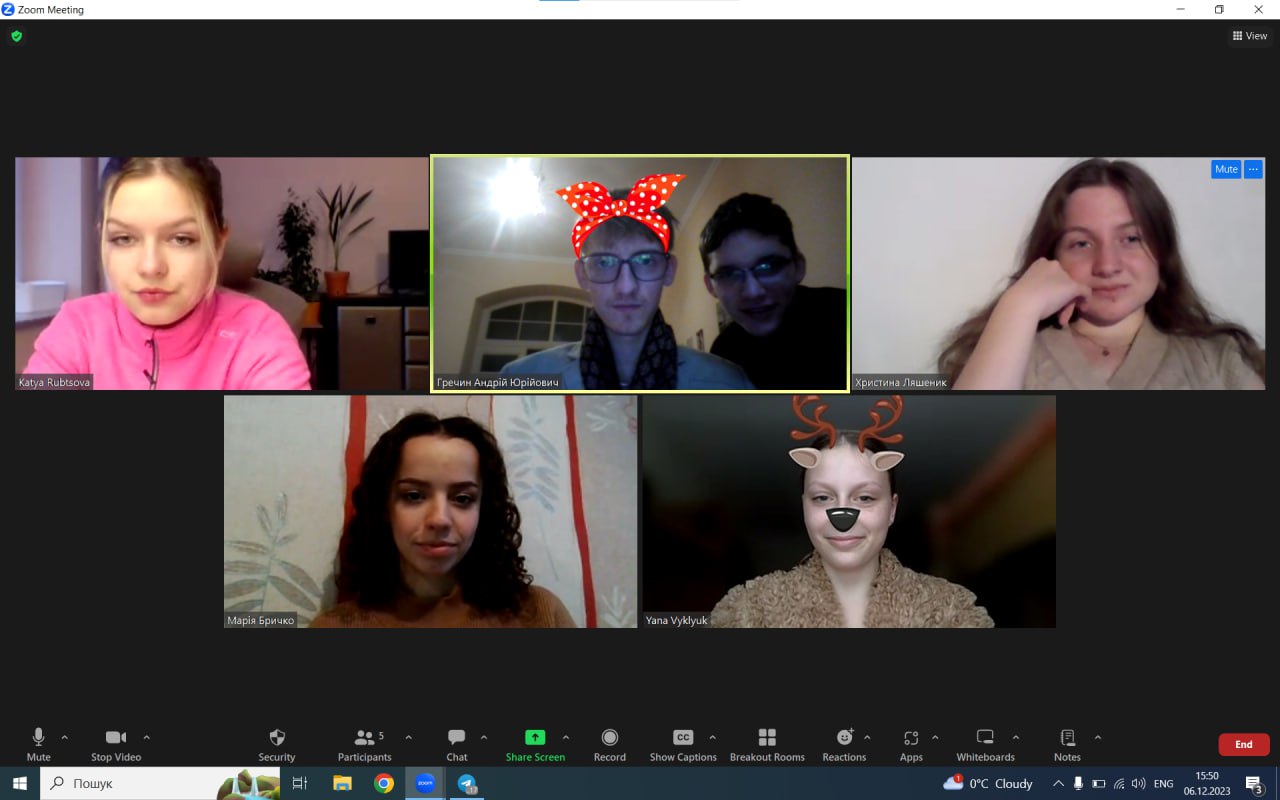


Блок з виконанням та тестуванням програми до завдання №9

Час затрачений на виконання завдання 40 хв.

## **6. Командна робота(мітинги, постановка задач та їх обговорення):**





# **Висновки:**

Освоєно навички роботи з файлами, системами числення, бінарними файлами, символами і рядковими змінними, а також текстовими файлами за допомогою стандартної бібліотеки. Отримано глибоке розуміння деталей і методів обробки файлів для ефективного використання та створення власних бібліотек.