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 »

***Виконав:***

студент групи ШІ-11

Винницький Андрій

# **Тема роботи:**

Опрацювання теми роботи з бінарними файлами, методами їх використання. Попрацював зі стрінгами. 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:**

Мета роботи передбачає вивчення та практичне опрацювання теми роботи з бінарними файлами та методів їх використання. Під час роботи проводилася робота з рядками, включаючи опрацювання стрінгів. Одним із важливих аспектів було написання кодів для вирішення практичних і лабораторних завдань з ВНС ,а також вирішення задач на платформі Algotester.

# **Теоретичні відомості:**

1. Індивідуальний план опрацювання теорії:

* Тема №1: Theory Education Activities
* <https://learn.microsoft.com/ru-ru/cpp/standard-library/fstream-typedefs?view=msvc-170>
* <https://cplusplus.com/reference/string/string/getline/>
* <https://ru.wikibooks.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%BE%D0%B2/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0/%D0%A8%D0%B5%D0%BB%D0%BB%D0%B0>
* <https://www.geeksforgeeks.org/shellsort/>
* <https://www.geeksforgeeks.org/insertion-sort/>
* <https://ppt-online.org/81798>
* <https://cplusplus.com/reference/string/string/>
* <https://cplusplus.com/reference/fstream/ifstream/>

Файли. Системи числення. Бінарні файли. Символи і рядкові змінні та текстові файли і деяка інформація для self practice.

* + Статус: Ознайомлений
  + 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: 9
* Деталі завдання: Надрукувати всі слова-паліндроми, які є в введеному рядку.
* Завдання №2: VNS Lab 8
* Варіант завдання: 9
* Деталі завдання
* Структура "Пацієнт":

- прізвище, ім'я, по батькові;

- домашня адреса;

- номер медичної карти;

- номер страхового поліса.

Знищити елемент із заданим номером медичної карти, додати 2 елементи в

початок файлу.

* Завдання №2: VNS Lab 9
* Варіант завдання: 9
* 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 містять тільки одне слово. Знайти найдовше слово у файлі F2.
* Завдання №3: Algotester Lab 4
* Варіант завдання: 3
* Деталі завдання: Вам дано масив, який складається з N додатніх цілих чисел. Ваше завдання - розділити його на три частини, по остачі від ділення на 3, по зростанню остачі (тобто спочатку йдуть числа, у яких остача 0, далі числа з остачею 1 і тоді нарешті числа з остачею 2). Далі необхідно ті елементи, остача від ділення на 3 яких парна посортувати по зростанню, а ті, у яких остача 1 - по спаданню. Після цього видаліть усі дублікати з масиву.
* Завдання №3: Algotester Lab 6
* Варіант завдання: 3
* Деталі завдання: У Клінта в черговий раз виключилось світло і йому немає чим зайнятися. Так як навіть це не заставить його подивитися збережені відео про програмування на ютубі - він вирішив придумати свою гру на основі судоку.

Гра виглядає так:

Є поле розміром N×N, в якому частина клітинок заповнена цифрами, а частина клітинок пусті(позначаються нулем). Також у нього є Q пар координат X та Y. Завданням гри є написати до кожної координати скільки чисел туди можна вписати (якщо вона пуста) і які це числа (обов’язково в посортовані по зростанню!). В клітинку можна вписати лише ті числа, які не зустрічаються в рядку та стовбці, які перетинаються у цій клітинці.

Під час гри поле не міняється!

Також необовязково, щоб це було валідне судоку! Якщо є клітинка, в яку не можна вписати ніяку цифру - виведіть 0.

Також допускаються рядки та стовпці, в яких цифра записана кілька разів.

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

* Реалізувати алгоритм сортування Shell(Sedgewick) sort, Insertion sort

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

Програма №1 VNS Lab 6

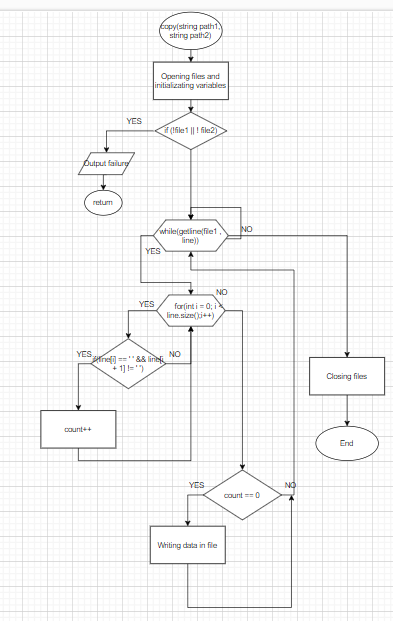
* Планований час на реалізацію 30 хв

Програма №2 VNS Lab 8

* Планований час на реалізацію 30 хв

Програма №3 VNS Lab 9

* Блок-схема



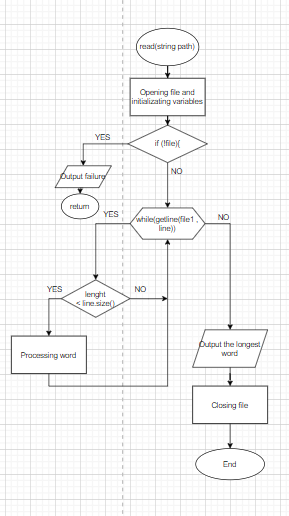
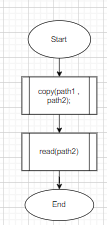
* 
* 

Figure 1:Блок-схема до задачі 9

* Планований час на реалізацію 30 хв

Програма №4 Algotester Lab 4

* Планований час на реалізацію 40 хв

Програма №5 Algotester Lab 6

* Планований час на реалізацію 15 хв

Програма №6 Class Practice Work

* Планований час на реалізацію 20 хв

Програма №7 Self Practice Work

Планований час на реалізацію 40 хв

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

Завдання №1 VNS Lab 6

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/744/files#diff-4ea045e86c3d8a286a82881f93f7daccc7581dede7111bd73ccfe38c2b78f9b5>

#include <iostream>  
#include<vector>  
using namespace std;  
  
  
vector<string> parse(string text)  
{  
 vector<string> data;  
 string buff;  
 for(int i = 0 ;i < text.size(); i++)  
 {  
 bool is\_separator\_found = false;  
 if (text[i] == ' ')  
 {  
 is\_separator\_found = true;  
 }  
  
 if (is\_separator\_found)  
 {  
 data.push\_back(buff);  
 buff = "";  
 }  
 else  
 {  
 if (text[i]!=' ')  
 {  
 buff += text[i];  
 }  
  
 }  
 }  
 return data;  
}  
bool is\_palindrome(string str, int first , int last )  
{  
 for (int k = 0; k <str.length(); k++)  
 {  
 str[k] = tolower(str[k]);  
 }  
 if(last > 0 && first < str.length())  
 {  
 if (str[first] == str[last]) {  
 return is\_palindrome(str, ++first, --last);  
 }  
 else  
 {  
 return false;  
 }  
  
 }  
 return true;  
  
  
}  
  
int main() {  
 string example ="level radar civic racecar noon elephant computer strawberry programming butterfly";  
 string str;  
 cout<<"Enter a string: ";  
 getline(cin , str);  
 cout<<"Entered a string: ";  
 cout<< str << endl;  
 vector<string> buff = parse(str);  
 cout<<"Palindrome words: ";  
 for (const auto & i : buff) {  
 if(is\_palindrome(i, 0, i.length() - 1))  
 {  
 cout<<i<<endl;  
 }  
 }  
 return 0;  
}

Завдання №2: VNS Lab 8

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/744/files#diff-20521c7fbfcd8fa3435e0b7aabad6167a3a3067e16998f800f1f24084668c4a2>

#include <iostream>  
#include <fstream>  
#include <vector>  
using namespace std;  
  
struct Patient  
{  
 string initials;  
 string adress;  
 string number\_of\_med;  
 string number\_of\_sec;  
};  
void deleteAllData(string path){  
 ofstream file(path, ios::trunc);  
 file.close();  
}  
void delete\_element(vector<Patient>& data, const string& number\_of\_cart)  
{  
 for(int i = 0; i < data.size(); i++)  
 if(data[i].number\_of\_med == number\_of\_cart)  
 {  
 data.erase(data.begin() + i);  
 }  
  
}  
vector<Patient> import\_data(vector<string> buffer)  
{  
  
 vector<Patient> data;  
 for (int i = 0; i < buffer.size() - 1; )  
 {  
 Patient buff = { buffer[i],buffer[i+1],buffer[i + 2],buffer[i + 3]};  
 i += 4;  
 data.push\_back(buff);  
 }  
 return data;  
}  
vector<string> parse(string text)  
{  
 vector<string> data;  
 string buff;  
 for(int i = 0 ;i < text.size(); i++)  
 {  
 bool is\_separator\_found = false;  
 if (text[i] == ';')  
 {  
 is\_separator\_found = true;  
 }  
  
 if (is\_separator\_found)  
 {  
 data.push\_back(buff);  
 buff = "";  
 }  
 else  
 {  
 if (text[i]!=' ')  
 {  
 buff += text[i];  
 }  
  
 }  
 }  
 return data;  
}  
  
string read(string path)  
{  
 string output;  
 string line;  
 ifstream file;  
 file.open(path, ios\_base::app);  
 if (!file){  
  
 return "Failure" ;  
  
 }  
 while(getline(file , line))  
 {  
 output += line;  
 cout<< line<<"\n";  
 }  
  
 file.close();  
 return output;  
}  
void write(string path, Patient data)  
{  
 ofstream file;  
 file.open(path,ios\_base::app);  
 if (!file){  
  
 cout<< "Failure" ;  
 return;  
  
 }  
 file << data.initials<<"; ";  
 file << data.adress<<"; ";  
 file << data.number\_of\_med<<"; ";  
 file << data.number\_of\_sec<<";\n ";  
 file.close();  
}  
  
  
  
int main()  
{  
  
 string path = "C:/Users/Nout\_1/CLionProjects/untitled3/data.txt";  
 write(path, {"Jonh Black","Chicago","7895","6547"});  
 write(path, {"Adam White","New York","5321","4596"});  
 write(path, {"Damian Brown","Minesota","6542","6548"});  
 cout<< "File's data:\n";  
 string text = read(path);  
 cout << "Enter a number of patient's medical cart: ";  
 string med\_card;  
 cin>> med\_card;  
 vector<Patient> data = import\_data(parse(text));  
 delete\_element(data, med\_card);  
 deleteAllData(path);  
 write(path, {"Mark Dominiko","Berlin","7895","6547"});  
 write(path, {"Volodymyr Duda","Lviv","5321","4596"});  
 for(int i = 0; i < data.size(); i++)  
 {  
 write(path,{data[i].initials,data[i].adress,data[i].number\_of\_med,data[i].number\_of\_sec});  
 }  
 cout<<"Edited list:\n ";  
 read(path);  
  
 return 0;  
}

Завдання №3: VNS Lab 9

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/744/files#diff-4251332441337ce6633c635d4d18f3fe673491b3746ade472e69caba1dd3e4b8>

#include <iostream>  
#include <fstream>  
#include <vector>  
using namespace std;  
  
void copy(string path1, string path2)  
{  
 string line;  
 ifstream file1;  
 ofstream file2;  
 file1.open(path1, ios\_base::app);  
 file2.open(path2, ios\_base::app);  
 if (!file1 || ! file2){  
  
 cout<<"Failure" ;  
 return;  
  
 }  
 while(getline(file1 , line))  
 {  
 int count = 0;  
 for(int i = 0; i < line.size();i++)  
 {  
 if(line[i] == ' ' && line[i + 1] != ' ')  
 {  
 count++;  
 }  
 }  
 if (count == 0)  
 {  
 file2<<line<<"\n";  
 }  
 }  
 file1.close();  
 file2.close();  
  
}  
void read(string path)  
{  
  
 string line;  
 ifstream file;  
 file.open(path, ios\_base::app);  
 if (!file){  
  
 cout<<"Failure" ;  
 return;  
  
 }  
 int lenght = 0;  
 string buff;  
 while(getline(file , line))  
 {  
 if (lenght < line.size())  
 {  
 lenght = line.size();  
 buff = line;  
 }  
  
 }  
 cout<<"The longest word: "<< buff;  
 file.close();  
  
}  
  
int main()  
{  
 string path1 = "C:/Users/Nout\_1/CLionProjects/untitled3/file\_1.txt";  
 string path2 = "C:/Users/Nout\_1/CLionProjects/untitled3/file\_2.txt";  
 copy(path1 , path2);  
 read(path2);  
  
 return 0;  
}

Завдання №4: Algotester Lab 4 v3 task1

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/744/files#diff-9c0ed31b4a33fd061757edf4f4de1227ebb0bbfad3b02a5720f0c9a4fab4fa6f>

#include <iostream>  
#include <vector>  
  
  
using namespace std;  
  
  
void merging(vector<int>& num\_arr, int start ,int midle, int end)  
{  
 vector<int> buff\_right(num\_arr.begin() + midle + 1, num\_arr.begin() + end + 1);  
 vector<int> buff\_left(num\_arr.begin() + start, num\_arr.begin() + midle + 1);  
 int ind = 0;  
 while (!buff\_left.empty() && !buff\_right.empty())  
 {  
 if (buff\_left[ind] > buff\_right[ind])  
 {  
 num\_arr[start] = buff\_right[ind];  
 buff\_right.erase(buff\_right.begin() + ind);  
 start++;  
 }  
 else  
 {  
 num\_arr[start] = buff\_left[ind];  
 buff\_left.erase(buff\_left.begin() + ind);  
 start++;  
 }  
  
 }  
 if(buff\_left.empty())  
 {  
 for(int i : buff\_right)  
 {  
 num\_arr[start] = i;  
 start++;  
 }  
  
 }  
 else if(buff\_right.empty())  
 {  
 for(int i : buff\_left)  
 {  
 num\_arr[start] = i;  
 start++;  
 }  
  
 }  
}  
  
void merge\_sort(vector<int>& num\_arr, int start , int end)  
{  
 if (end != start)  
 {  
 int mid = (end + start)/2;  
  
 merge\_sort(num\_arr, start , mid);  
 merge\_sort(num\_arr, mid + 1 ,end);  
  
 merging(num\_arr, start, mid, end);  
 }  
  
}  
void reverse(vector<int>& num\_arr)  
{  
 for(int i = 0;i < num\_arr.size() / 2;i++)  
 {  
 int buff = num\_arr[i];  
 num\_arr[i] = num\_arr[num\_arr.size() - 1 - i];  
 num\_arr[num\_arr.size() - 1 - i] = buff;  
 }  
  
}  
  
  
  
int main(){  
 int size;  
 cin >> size;  
 vector<int> numbers;  
 for (int i = 0; i < size; i++){  
 int number;  
 cin >> number;  
 numbers.push\_back(number);  
 }  
 vector <int> buff0, buff1, buff2;  
 for (int i = 0; i < size; i++){  
 if (numbers[i] % 3 == 0){  
 buff0.push\_back(numbers[i]);  
 }  
 else if (numbers[i] % 3 == 1){  
 buff1.push\_back(numbers[i]);  
 }  
 else if (numbers[i] % 3 == 2){  
 buff2.push\_back(numbers[i]);  
 }  
 }  
  
 merge\_sort(buff0, 0, buff0.size() - 1);  
 merge\_sort(buff1, 0, buff1.size() - 1);  
  
 reverse(buff1);  
 merge\_sort(buff2, 0, buff2.size() - 1);  
 buff0.insert(buff0.end(), buff1.begin(), buff1.end());  
 buff0.insert(buff0.end(), buff2.begin(), buff2.end());  
  
 vector<int> buff;  
 for(int i = 0;i < buff0.size() ;i++)  
 {  
  
 if (i == 0 || buff0[i] != buff0[i - 1])  
 {  
 buff.push\_back(buff0[i]);  
 }  
  
 }  
 cout<<buff.size()<<endl;  
 for(int i : buff)  
 {  
 cout<<i<<' ';  
 }  
  
 return 0;  
}

Завдання №4: Algotester Lab 4 v3 task 2

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/744/files#diff-f1cad5affdaabf0cabf4d1b3db9f1a1fbed00cd2b57966673005a9a2f0e04e0d>

#include <iostream>  
#include <vector>  
#include <algorithm>  
  
using namespace std;  
  
int main(){  
 int size;  
 cin >> size;  
 vector<int> numbers;  
 for (int i = 0; i < size; i++){  
 int number;  
 cin >> number;  
 numbers.push\_back(number);  
 }  
 vector <int> buff0, buff1, buff2;  
 for (int i = 0; i < size; i++){  
 if (numbers[i] % 3 == 0){  
 buff0.push\_back(numbers[i]);  
 }  
 else if (numbers[i] % 3 == 1){  
 buff1.push\_back(numbers[i]);  
 }  
 else if (numbers[i] % 3 == 2){  
 buff2.push\_back(numbers[i]);  
 }  
 }  
  
 sort(buff0.begin(),buff0.end());  
 sort(buff1.begin(),buff1.end());  
 reverse(buff1.begin(), buff1.end());  
 sort(buff2.begin(),buff2.end());  
 buff0.insert(buff0.end(), buff1.begin(), buff1.end());  
 buff0.insert(buff0.end(), buff2.begin(), buff2.end());  
  
 buff0.erase(unique(buff0.begin(), buff0.end()), buff0.end());  
  
 cout<<buff0.size()<<endl;  
 for(int i : buff0)  
 {  
 cout<<i<<' ';  
 }  
  
 return 0;  
}

Завдання №5: Algotester Lab 6 v3

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/744/files#diff-bfb822267832eb49614ef4125627f83a29dd34a49a09bf9ca624cf857ff6d7d8>

#include <iostream>  
#include <vector>  
using namespace std;  
  
void check(vector<string> arr , int value1 , int value2 )  
{  
  
 if (arr[value1][value2] != '0')  
 {  
 cout<<1<<endl<<arr[value1][value2]<<endl<<endl;  
 return;  
 }  
 vector<char> right;  
 for(int f = 1;f <arr[value1].size() + 1; f++)  
 {  
 right.push\_back(f + 48);  
 }  
 vector<char> buff;  
 for(char i : arr[value1])  
 {  
 if(i!='0')  
 {  
 buff.push\_back(i);  
 }  
 }  
 for(int l = 0;l <arr[value1].size(); l++)  
 {  
 if(arr[l][value2]!='0')  
 {  
 buff.push\_back(arr[l][value2]);  
 }  
 }  
 for(char k : buff)  
 {  
 for(int h = 0;h <right.size(); h++)  
 {  
 if(k == right[h])  
 {  
 right.erase(right.begin() + h);  
 }  
 }  
 }  
 cout << right.size() << endl;  
 for(char a : right)  
 {  
 cout<<a<<" ";  
 }  
 cout<<endl<<endl;  
}  
  
  
  
int main(){  
 int count;  
 cin>>count;  
 vector<string> arr;  
 for(int i = 0; i < count; i++)  
 {  
 string value;  
 cin>> value;  
 arr.push\_back(value);  
 }  
 int count\_a;  
 cin>>count\_a;  
 for(int j = 0; j < count\_a; j++)  
 {  
 int value1 , value2;  
 cin>> value1>>value2;  
 check(arr, value1 - 1, value2 - 1);  
  
 }  
 return 0;  
}

Завдання №5: Class Practice Task

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/744/files#diff-22ac74d50daaf7473165fe8c6c6884eba779a82de5dea362f1bc56bb9138713c>

#include<iostream>  
#include<fstream>  
#include <vector>  
using namespace std;  
  
vector<string> parse(string text)  
{  
 vector<string> data;  
 string buff;  
 for(int i = 0 ;i < text.size(); i++)  
 {  
 bool is\_separator\_found = false;  
 if (text[i] == ';')  
 {  
 is\_separator\_found = true;  
 }  
  
 if (is\_separator\_found)  
 {  
 data.push\_back(buff);  
 buff = "";  
 }  
 else  
 {  
  
 buff += text[i];  
  
  
 }  
 }  
 return data;  
}  
  
string read(string path)  
{  
 string output;  
 string line;  
 ifstream file;  
 file.open(path, ios\_base::app);  
 if (!file){  
  
 return "Failure" ;  
 }  
 cout << "Success" << endl;  
 while(getline(file , line))  
 {  
 output += line;  
 cout<< line<<"\n";  
 }  
  
 file.close();  
 return output;  
}  
void write(string path, string data)  
{  
  
 ofstream file;  
  
 file.open(path,ios\_base::app);  
 if (!file){  
  
 cout<< "Failure" ;  
 return;  
 }  
 cout << "Success" << endl;  
  
 file << data <<";\n";  
 file.close();  
}  
void deleteAllData(string path){  
 ofstream file(path, ios::trunc);  
 file.close();  
}  
  
int main()  
{  
 vector <string> sentences = **{**"The quick brown fox jumps over the lazy dog.", "Artificial intelligence is transforming various industries.","Coding allows us to create incredible software applications.","The sun sets in the west, painting the sky with vibrant colors.","Learning new things is a lifelong adventure."**}**;  
  
 string path = "C:/Users/Nout\_1/CLionProjects/untitled3/write.txt";  
 string path1 = "C:/Users/Nout\_1/CLionProjects/untitled3/copy.txt";  
 for (string i: sentences) {  
 write(path, i);  
 }  
  
 string rfile = read(path);  
 vector<string> r\_file = parse(rfile);  
 for (string i: r\_file) {  
 write(path1, i);  
 }  
   
 return 0;  
}

Завдання №6: Self Practice Task

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/744/files#diff-f6b3bacc6a1a24b403db51fe380f6f1cbc5963aa449825997add69e7392994c5>

#include <iostream>  
#include <vector>  
#include <ctime>  
#include <cmath>  
  
using namespace std;  
void print\_array(vector<int> num, int start , int end)  
{  
 for(int i = start; i < end; i++)  
 {  
 cout << num[i]<<" ";  
 }  
}  
  
void Shell\_sort\_sedgvik(vector<int> numbers, vector<int>key)  
{  
 int counter = 0;  
 for(int k = key.size() - 1; k >= 0; k--) {  
 int buff;  
  
 for (int i = key[k]; i < numbers.size(); i ++) {  
 buff = numbers[i];  
 counter++;  
 int ind = i - key[k];  
 for (; ind >= 0 && numbers[ind] > buff; ind-=key[k]) {  
 counter++;  
 numbers[ind + key[k]] = numbers[ind];  
 }  
 numbers[ind + key[k]] = buff;  
  
 }  
 }  
  
 cout<<"\nSorted array: ";  
 print\_array(numbers, 0, numbers.size());  
  
 cout<<"\nNumber of iterations: "<<counter;  
}  
void insertion\_sort(vector<int> numbers)  
{ long long int counter = 0;  
 int buff;  
 for(int i = 0; i < numbers.size(); i++)  
 {  
 counter++;  
 buff = numbers[i];  
 int ind = i - 1;  
 for(;ind >= 0 && numbers[ind] > buff;ind--)  
 {  
 counter++;  
 numbers[ind + 1] = numbers[ind];  
 }  
 numbers[ind + 1] = buff;  
  
 }  
  
 cout<<"\nSorted array: ";  
 print\_array(numbers, 0, numbers.size());  
  
 cout<<"\nNumber of iterations: "<<counter;  
}  
void Shell\_sort(vector<int> numbers)  
{ int counter = 0;  
 for(int k = numbers.size(); k > 0; k/=2) {  
 int buff;  
  
 for (int i = k; i < numbers.size(); i ++) {  
 counter++;  
 buff = numbers[i];  
 int ind = i - k;  
 for (; ind >= 0 && numbers[ind] > buff; ind-=k) {  
 counter++;  
 numbers[ind + k] = numbers[ind];  
 }  
 numbers[ind + k] = buff;  
  
 }  
 }  
  
 cout<<"\nSorted array: ";  
 print\_array(numbers, 0, numbers.size());  
  
 cout<<"\nNumber of iterations: "<<counter;  
}  
  
  
vector<int> Sedgvik(int size)  
{  
 vector<int>steps;  
 int count = 0;  
 do{  
 if (count % 2 != 0) {  
 int buff = 8 \* pow(2,count) - 6\* pow(2, (count + 1)/2) + 1;  
 steps.push\_back(buff);  
 count++;  
 }  
 else {  
 int buff1 = 9 \* pow(2,count) - 9 \* pow(2,count/2) + 1;  
 steps.push\_back(buff1);  
 count++;  
 }  
 }while(3\*steps[count - 1] < size);  
 steps.erase(steps.end() - 1);  
 return steps;  
}  
  
int main()  
{  
 srand(time(0));  
 vector<int> nums;  
 for(int i = 0; i < 100;i++)  
 {  
 nums.push\_back(rand() % 100 );  
 }  
 cout<<"\nUnSorted array: ";  
 print\_array(nums, 0, nums.size());  
 vector<int> buff = Sedgvik(nums.size());  
  
 Shell\_sort\_sedgvik(nums, buff);  
  
 Shell\_sort(nums);  
  
 insertion\_sort(nums);  
  
 return 0;  
}

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

Завдання №VNS Lab 6

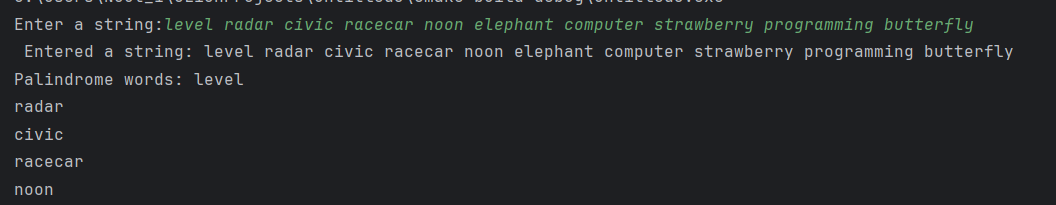


Figure 2:Програма опрацьовує рядок данних і виводить слова паліндроми

Час затрачений на виконання завдання: 20 хв.

Завдання №2: VNS Lab 8

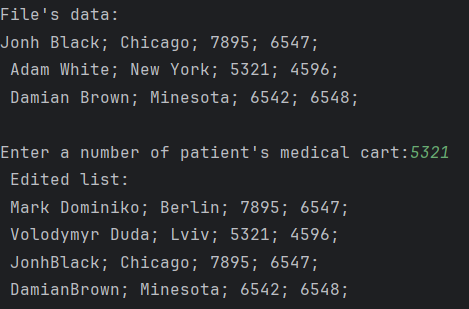


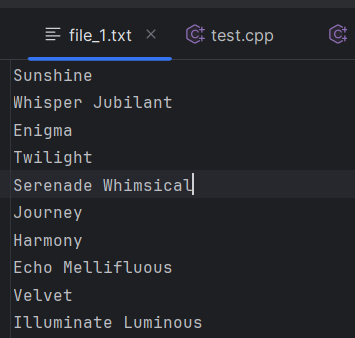
Figure 3: програма читає дані з файлу опрацьовує їх , виводить і проводить над ними потрібні операції

Час затрачений на виконання завдання: 3 год.

Завдання №2: VNS Lab 9



Figure 4: Програма копією дані за певною умовою в інший файл і з тих слів друкує найдовше



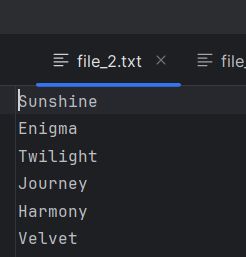


Figure 5: Файли з даними

Час затрачений на виконання завдання: 30 хв.

Завдання №3: Algotester Lab 4 using own sorting

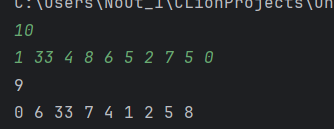
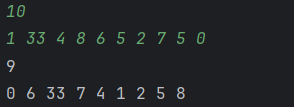




Figure 6:Результат коду на алготестері

Час затрачений на виконання завдання: 1 год

Завдання №3: Algotester Lab 4 using std methods



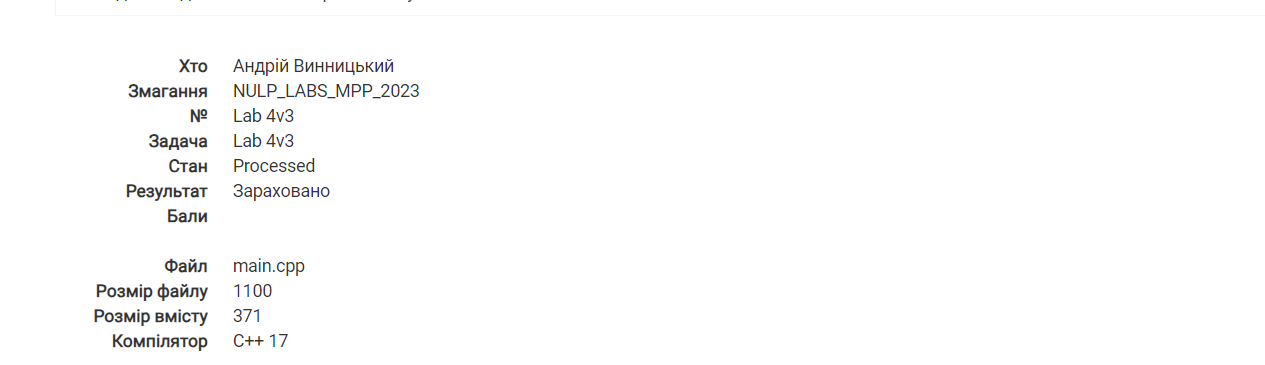


Figure 7: Результат коду на алготестері

Завдання №4: Algotester Lab 6

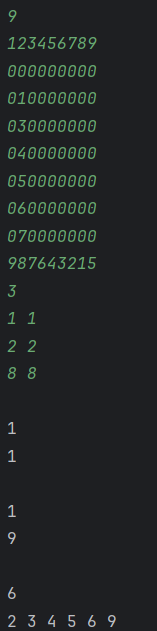


Figure 8:Результат коду



Figure 9:Результат коду на алготестері

Час затрачений на виконання завдання: 20 хв

Завдання №5: Class Practice Task

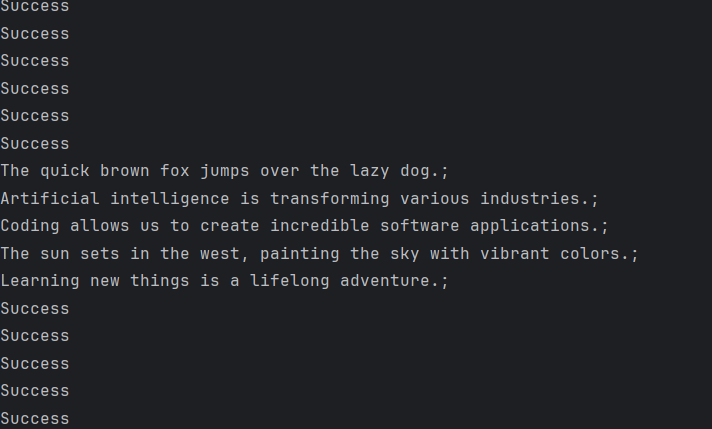
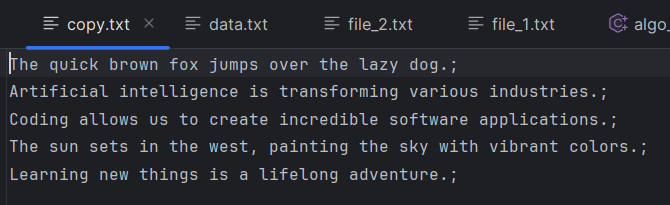


Figure 10: Програма записує дані в один файл і копіює в інший



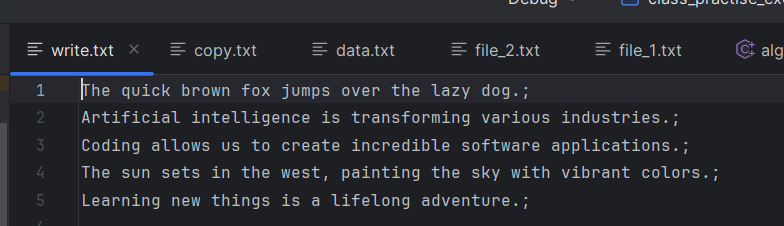


Figure 21:Файли програми

Час затрачений на виконання завдання: 30 хв

Завдання №6: Self Practice Task

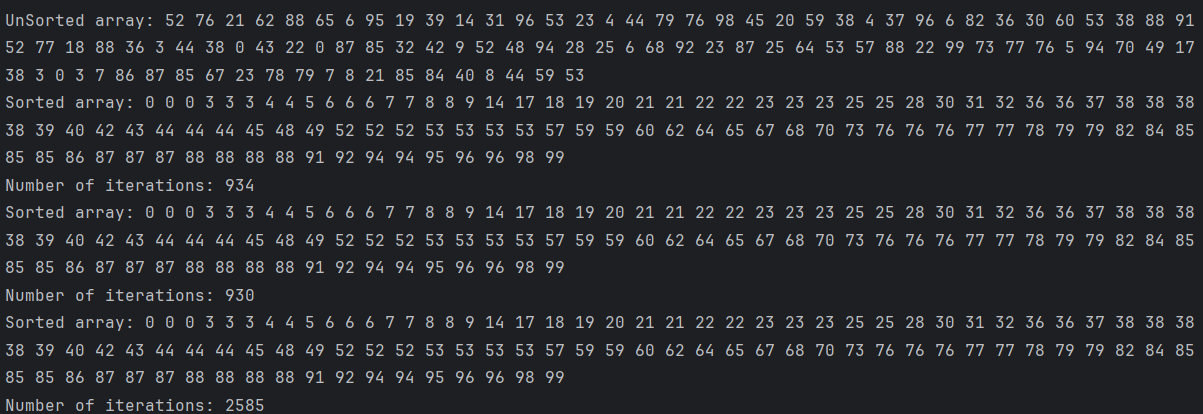


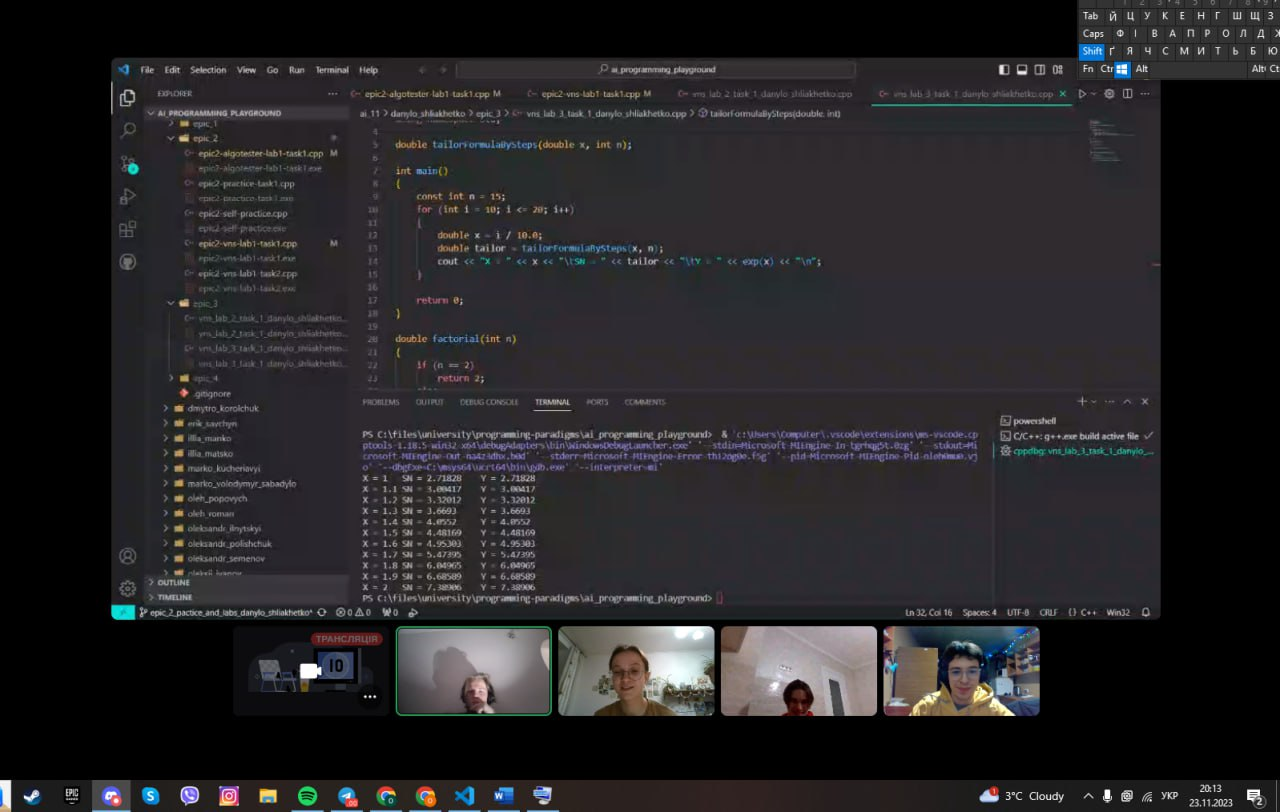
Figure 3:Програма сортує массив 3 різними алгоритмами

Час затрачений на виконання завдання: 5 год

## **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 даній лабораторній роботі я освоїв поняття бінарних файлів, роботу з текстовими файлами та методи їх використання. Крім цього, я ознайомився із стандартними бібліотеками та вивчив їх застосування. Для закріплення отриманих знань нам було запропоновано вирішити декілька завдань з ВНС та виконати задачі на платформі Algotester. Усі необхідні файли, включаючи звіт, я успішно завантажив на GitHub.