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-11

Шляхетко Данило Віталійович

# **Тема роботи:**

Знайомство з файлами, системами числення, бінарними файлами, символами і рядковими змінними та текстовими файлами, стандартною бібліотекою та методами роботи з файлами, створенням й використанням бібліотек.

# **Мета роботи:**

Ознайомитись з файлами, системами числення, бінарними файлами, символами і рядковими змінними та текстовими файлами, стандартною бібліотекою та методами роботи з файлами, створенням й використанням бібліотек.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Символи і рядкові змінні.
* Тема №2: Файли.
* Тема №3: Стандартна бібліотека.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Символи і рядкові змінні.
  + Джерела Інформації
    - Лекції та практичні
    - Попередній досвід
  + Що опрацьовано:
    - Як працювати з символи і рядковими змінними
  + Статус: Ознайомлений
  + Початок опрацювання теми: 09/12/2023
  + Звершення опрацювання теми: 03/12/2023
* Тема №2: Файли.
  + Джерела Інформації
    - Лекції та практичні
    - Попередній досвід
  + Що опрацьовано:
    - Як працювати з файлами в C++
  + Статус: Ознайомлений
  + Початок опрацювання теми: 09/12/2023
  + Звершення опрацювання теми: 03/12/2023
* Тема №3: Стандартна бібліотека.
  + Джерела Інформації
    - Лекції та практичні
    - Попередній досвід
  + Що опрацьовано:
    - Як працювати з стандартною бібліотекою C++
  + Статус: Ознайомлений
  + Початок опрацювання теми: 09/12/2023
  + Звершення опрацювання теми: 03/12/2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 6

* Варіант 5
* Перетворити рядок таким чином, щоб спочатку в ньому були надруковані тільки букви, а потім тільки цифри, не міняючи порядку проходження символів у рядку.

Завдання №2 VNS Lab 8

* Варіант 5
* 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 "Людина": - прізвище, ім'я, по батькові; - рік народження; - ріст; - вага. Знищити усі елементи із зазначеним ростом і вагою, додати елемент після елемента із зазначеним прізвищем.

Завдання №3 VNS Lab 9

* Варіант 5
* Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію. 1) Скопіювати з файлу F1 у файл F2 рядки, починаючи з K до K+5. 2) Підрахувати кількість голосних букв у файлі F2.

Завдання №4 Algotester Lab 4

* Варіант 3
* 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. Виведіть результуючий масив.

Завдання №5 Algotester Lab 6

* Варіант 3
* 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 №6 Class Practice Task

* 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.

* Реалізувати функцію створення файла і запису в нього даних:

enum FileOpResult { Success, Failure, … };

FileOpResult copy\_file(char \*file\_from, char \*file\_to);

Умови задачі:

- копіювати вміст файла з ім’ям file\_from у файл з ім’ям file\_to; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів

- file\_from, file\_to – можуть бути повним або відносним шляхом

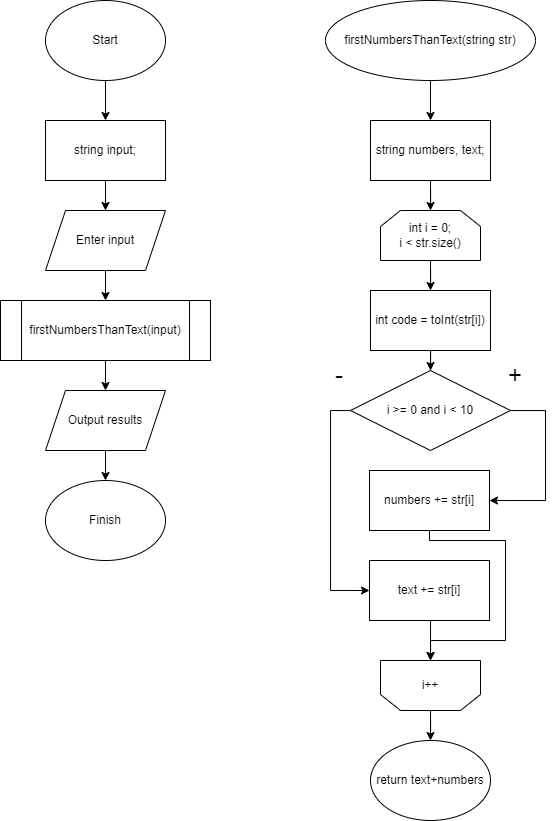
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файла.

Завдання №7 Self Practice Task

* Написати програму з використанням нового матеріалу

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

Завдання №1 VNS Lab 6



*Малюнок 1. VNS Lab 6*

* Запланований час на виконання: 30 хв.

Завдання №2 VNS Lab 8

* Запланований час на виконання: 1 год.

Завдання №3 VNS Lab 9

* Запланований час на виконання: 1 год.

Завдання №4 Algotester Lab 4

* Запланований час на виконання: 30 хв.

Завдання №5 Algotester Lab 6

* Запланований час на виконання: 1 год.

Завдання №6 Class Practice Task

* Запланований час на виконання: 30 хв.

Завдання №7 Self Practice Task

* Запланований час на виконання: 30 хв.

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

Завдання №1 VNS Lab 6

* <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/869/files#diff-933983bbb7add466d85fdfeb9d86b7bd13e371026ba3ead699834a8ea479f781>
* #include <iostream>
* #include <string>
* using namespace std;
* string firstNumbersThanText(string str)
* {
* string numbers = "";
* string text = "";
* for (int i = 0; i < str.size(); i++)
* {
* int code = int(str[i]) - 48;
* if (code >= 0 && code < 10)
* {
* numbers += str[i];
* }
* else
* {
* text += str[i];
* }
* }
* return text + numbers;
* }
* int main()
* {
* string input;
* cin >> input;
* cout << firstNumbersThanText(input);
* return 0;
* }

Завдання №2 VNS Lab 8

* <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/869/files#diff-30a13c611a4db325dc94003ab74076f2e8c8ec5c20c428784fc00d05b4be81c0>

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

struct Human

{

string personInitials;

string yearOfBirth;

string height;

string weight;

};

vector<string> parseStringToStringArr(string input)

{

vector<string> arr;

string buff;

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

{

bool isSeparator = false;

if (input[i] == ';')

{

isSeparator = true;

}

if (isSeparator)

{

arr.push\_back(buff);

buff = "";

}

else

{

buff += input[i];

}

}

return arr;

}

vector<Human> parseStringArrToHumanArr(vector<string> parsedData)

{

vector<Human> data;

for (int i = 0; i < parsedData.size() - 1; i += 4)

{

Human buff = {parsedData[i], parsedData[i + 1], parsedData[i + 2], parsedData[i + 3]};

data.push\_back(buff);

}

return data;

}

vector<Human> deleteElementFromArr(vector<Human> arr, string height, string weight)

{

vector<Human> newArr;

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

{

if (arr[i].height != height && arr[i].weight != weight)

{

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

}

}

return newArr;

}

void writeLine(string path, Human data)

{

ofstream file;

file.open(path, ios\_base::app);

if (!file)

{

cout << "File doesn't exist\n";

return;

}

file << data.personInitials << ";"

<< data.yearOfBirth << ";"

<< data.height << ";"

<< data.weight << ";\n";

file.close();

}

string readAll(string path)

{

string output;

string currentLine;

ifstream file;

file.open(path, ios\_base::app);

if (!file)

{

return "File doesn't exist\n";

}

while (getline(file, currentLine))

{

output += currentLine;

cout << currentLine << "\n";

}

file.close();

return output;

}

void deleteFileContents(string path)

{

ofstream file(path, ios::trunc);

file.close();

}

const string PATH = "C:/files/university/programming-paradigms/data/data.txt";

int main()

{

deleteFileContents(PATH);

writeLine(PATH, {"Shliakhetko Danylo Vitaliovych", "2006", "188", "72"});

writeLine(PATH, {"Vinnyk Andrew Olehovych", "2007", "185", "67"});

writeLine(PATH, {"Chuma David Serhiyovych", "2005", "180", "75"});

writeLine(PATH, {"Pediak Uliana Viktorivna", "2002", "174", "50"});

writeLine(PATH, {"Kozlovskyi Anatolii Stepanovych", "2000", "195", "120"});

cout << "Created file contents:\n";

string contents = readAll(PATH);

vector<Human> arr = parseStringArrToHumanArr(parseStringToStringArr(contents));

cout << "Enter person's height to delete: ";

string height;

cin >> height;

cout << "Enter person's weight to delete: ";

string weight;

cin >> weight;

arr = deleteElementFromArr(arr, height, weight);

deleteFileContents(PATH);

writeLine(PATH, {"Sabdyliuk Marko Vasylovych", "2004", "174", "62"});

writeLine(PATH, {"Zinoviev Volodymyr Alexandrovych", "2003", "190", "80"});

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

{

writeLine(PATH, {arr[i].personInitials, arr[i].yearOfBirth, arr[i].height, arr[i].weight});

}

cout << "New file contents:\n";

readAll(PATH);

return 0;

}

Завдання №3 VNS Lab 9

* <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/869/files#diff-be0a59a7d42bc7e49c6bf93c55c72a4616cb339058a30fae76906ffe360c170b>

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

const string PATH = "C:/files/university/programming-paradigms/data/";

const string F1 = "F1.txt";

const string F2 = "F2.txt";

void read(string path)

{

ifstream file;

file.open(path, ios\_base::app);

}

void copyF1ToF2(string path, int k)

{

int max = k + 5;

if (k < 0 || max > 10)

{

cout << "Enter valid k";

return;

}

vector<string> output;

string currentLine;

ifstream readFile;

readFile.open(path + F1, ios\_base::app);

if (!readFile)

{

cout << "File doesn't exist\n";

return;

}

cout << "Reading file F1...\n\n";

while (getline(readFile, currentLine))

{

cout << currentLine << "\n";

output.push\_back(currentLine);

}

readFile.close();

cout << "\nRead file F1"

<< "\n";

ofstream deleteFile(path + F2, ios::trunc);

deleteFile.close();

ofstream writeFile;

writeFile.open(path + F2, ios\_base::app);

if (!writeFile)

{

cout << "File doesn't exist\n";

return;

}

for (; k < max; k++)

{

writeFile << output[k] << "\n";

}

writeFile.close();

cout << "Write file F2"

<< "\n\n";

}

void countVowels(string path)

{

string output;

string currentLine;

ifstream readFile;

readFile.open(path + F2, ios\_base::app);

if (!readFile)

{

cout << "File doesn't exist\n";

return;

}

cout << "Reading file F2...\n\n";

while (getline(readFile, currentLine))

{

cout << currentLine << "\n";

output += currentLine;

}

readFile.close();

cout << "\nRead file F2"

<< "\n";

int count = 0;

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

{

char c = output[i];

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

{

count++;

}

}

cout << "In file F2 there are " << count

<< " vowels\n";

}

int main()

{

int input;

cout << "Enter k:\n";

cin >> input;

copyF1ToF2(PATH, input);

countVowels(PATH);

return 0;

}

Завдання №4 Algotester Lab 4

* Task 1
* <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/869/files#diff-81f3eadebd195aa402da080f9f7849d24766e3f05f78115202aa40c3c08b6edb>
* #include <iostream>
* #include <vector>
* #include <algorithm>
* using namespace std;
* int main()
* {
* int n;
* cin >> n;
* vector<int> arr0;
* vector<int> arr1;
* vector<int> arr2;
* for (int i = 0; i < n; i++)
* {
* int input;
* cin >> input;
* if (input % 3 == 0)
* {
* arr0.push\_back(input);
* }
* else if (input % 3 == 1)
* {
* arr1.push\_back(input);
* }
* else
* {
* arr2.push\_back(input);
* }
* }
* sort(arr0.begin(), arr0.end());
* sort(arr1.begin(), arr1.end(), greater<int>());
* sort(arr2.begin(), arr2.end());
* arr0.erase(unique(arr0.begin(), arr0.end()), arr0.end());
* arr1.erase(unique(arr1.begin(), arr1.end()), arr1.end());
* arr2.erase(unique(arr2.begin(), arr2.end()), arr2.end());
* cout << arr0.size() + arr1.size() + arr2.size() << "\n";
* for (int i = 0; i < arr0.size(); i++)
* {
* cout << arr0[i] << " ";
* }
* for (int i = 0; i < arr1.size(); i++)
* {
* cout << arr1[i] << " ";
* }
* for (int i = 0; i < arr2.size(); i++)
* {
* cout << arr2[i] << " ";
* }
* return 0;
* }
* Task 2
* <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/869/files#diff-b5b765d39e2de384093eeec80a386902e95be4ff65f95e5bb4e971969eed9a9c>

#include <iostream>

#include <vector>

using namespace std;

int partition(vector<int> &arr, int start, int end)

{

int currentElement = arr[end];

int i = start - 1;

for (int j = start; j < end; j++)

{

if (arr[j] <= currentElement)

{

i++;

swap(arr[i], arr[j]);

}

}

swap(arr[i + 1], arr[end]);

return i + 1;

}

void quickSort(vector<int> &arr, int start, int end)

{

if (start < end)

{

int currentElementIndex = partition(arr, start, end);

quickSort(arr, start, currentElementIndex - 1);

quickSort(arr, currentElementIndex + 1, end);

}

}

void quickSort(vector<int> &arr)

{

quickSort(arr, 0, arr.size() - 1);

}

void reverse(vector<int> &arr, int start, int end)

{

if (start > end)

return;

swap(arr[start], arr[end]);

reverse(arr, start + 1, end - 1);

}

void reverse(vector<int> &arr)

{

reverse(arr, 0, arr.size() - 1);

}

void removeRepeatableElements(vector<int> &arr)

{

vector<int> newArr;

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

{

Repeat:

if (arr[i] == arr[i + 1] && i < arr.size()-1)

{

i++;

goto Repeat;

}

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

}

arr = newArr;

}

int main()

{

int n;

cin >> n;

vector<int> arr0;

vector<int> arr1;

vector<int> arr2;

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

{

int input;

cin >> input;

if (input % 3 == 0)

{

arr0.push\_back(input);

}

else if (input % 3 == 1)

{

arr1.push\_back(input);

}

else

{

arr2.push\_back(input);

}

}

quickSort(arr0, 0, arr0.size() - 1);

quickSort(arr1, 0, arr1.size() - 1);

reverse(arr1);

quickSort(arr2, 0, arr2.size() - 1);

removeRepeatableElements(arr0);

removeRepeatableElements(arr1);

removeRepeatableElements(arr2);

cout << arr0.size() + arr1.size() + arr2.size() << "\n";

for (int i = 0; i < arr0.size(); i++)

{

cout << arr0[i] << " ";

}

for (int i = 0; i < arr1.size(); i++)

{

cout << arr1[i] << " ";

}

for (int i = 0; i < arr2.size(); i++)

{

cout << arr2[i] << " ";

}

return 0;

}

Завдання №5 Algotester Lab 6

* <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/869/files#diff-9734bd7dce70620bdcd756d2bd482b3454aafc64fcb2b937c7eb3a142741acd1>

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

vector<int> getPossibleValues(vector<vector<int>> fields, int first, int second)

{

int size = fields.size();

vector<int> result;

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

{

result.push\_back(i);

}

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

{

auto answer = find(result.begin(), result.end(), fields[i][second]);

if (answer != result.end())

{

result.erase(answer);

}

}

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

{

auto answer = find(result.begin(), result.end(), fields[first][i]);

if (answer != result.end())

{

result.erase(answer);

}

}

return result;

}

int main()

{

int size;

cin >> size;

vector<vector<int>> fields;

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

{

vector<int> currentArr;

string line;

cin >> line;

for (int j = 0; j < line.size(); j++)

{

int currentNumber = int(line[j]) - 48;

currentArr.push\_back(currentNumber);

}

fields.push\_back(currentArr);

}

int coordinatesSize;

cin >> coordinatesSize;

vector<pair<int, int>> coordinates;

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

{

int x, y;

cin >> x >> y;

pair<int, int> currentPair = make\_pair(x - 1, y - 1);

coordinates.push\_back(currentPair);

}

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

{

if (fields[coordinates[i].first][coordinates[i].second] != 0)

{

cout << 1 << "\n"

<< fields[coordinates[i].first][coordinates[i].second] << "\n\n";

}

else

{

vector<int> possibleValues = getPossibleValues(fields, coordinates[i].first, coordinates[i].second);

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

for (int j = 0; j < possibleValues.size(); j++)

{

cout << possibleValues[j] << " ";

}

cout << "\n\n";

}

}

return 0;

}

Завдання №6 Class Practice Task

* <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/869/files#diff-7d0a7bfb7dddcfb644932ad017574e225091057d4b55f27817db1615a309bc67>

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

using namespace std;

const string PATH = "C:/files/university/programming-paradigms/data/";

const string FILE\_FROM = "file\_from.txt";

const string FILE\_TO = "file\_to.txt";

void createFile(string path, string content)

{

ofstream deleteFile(path + ".txt", ios::trunc);

deleteFile.close();

ofstream writeFile;

writeFile.open(path + ".txt", ios\_base::app);

if (!writeFile)

{

cout << "Failure\n";

return;

}

writeFile << content;

writeFile.close();

cout << "Success\n";

}

void copyFile(string fromPath, string toPath)

{

vector<string> output;

string currentLine;

ifstream readFile;

readFile.open(fromPath, ios\_base::app);

if (!readFile)

{

cout << "Failure\n";

return;

}

while (getline(readFile, currentLine))

{

cout << currentLine << "\n";

output.push\_back(currentLine);

}

readFile.close();

ofstream deleteFile(toPath, ios::trunc);

deleteFile.close();

ofstream writeFile;

writeFile.open(toPath, ios\_base::app);

if (!writeFile)

{

cout << "Failure\n";

return;

}

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

{

writeFile << output[i] << "\n";

}

writeFile.close();

cout << "Success\n";

}

int main()

{

cout << "--- Write File ---\n";

cout << "Enter file name:\n";

string name;

cin >> name;

cout << "Enter file content:\n";

string content;

cin >> content;

createFile(PATH + name, content);

cout << "--- Copy File ---\n";

copyFile(PATH + FILE\_FROM, PATH + FILE\_TO);

return 0;

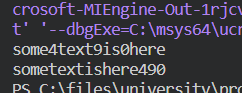
}

Завдання №7 Self Practice Task

* <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/869/files#diff-dc247899b711997c5abd2a8e0fe5b18b3e7c5357ed6ab6868224a5e0760c061b>
* #include <iostream>
* #include <vector>
* using namespace std;
* int partition(vector<int> &arr, int start, int end)
* {
* int currentElement = arr[end];
* int i = start - 1;
* for (int j = start; j < end; j++)
* {
* if (arr[j] <= currentElement)
* {
* i++;
* swap(arr[i], arr[j]);
* }
* }
* swap(arr[i + 1], arr[end]);
* return i + 1;
* }
* vector<int> quickSort(vector<int> &arr, int start, int end)
* {
* if (start < end)
* {
* int currentElementIndex = partition(arr, start, end);
* quickSort(arr, start, currentElementIndex - 1);
* quickSort(arr, currentElementIndex + 1, end);
* }
* return arr;
* }
* vector<int> quickSort(vector<int> &arr)
* {
* return quickSort(arr, 0, arr.size() - 1);
* }
* int main()
* {
* vector<int> arr = {5, 6, 3, 9, 1, 4, 2, 8, 7, 0};
* vector<int> sortedArr = quickSort(arr);
* for (int i = 0; i < sortedArr.size(); i++)
* {
* cout << sortedArr[i] << " ";
* }
* return 0;
* }

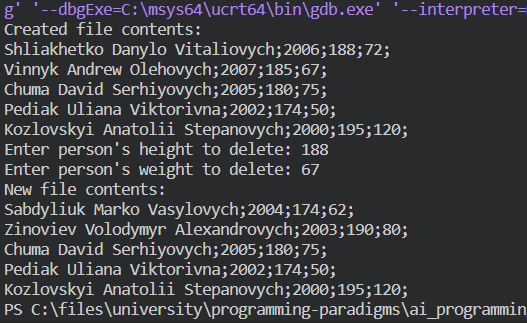
## **4. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 VNS Lab 6

* Затрачено: 20 хв.
* 

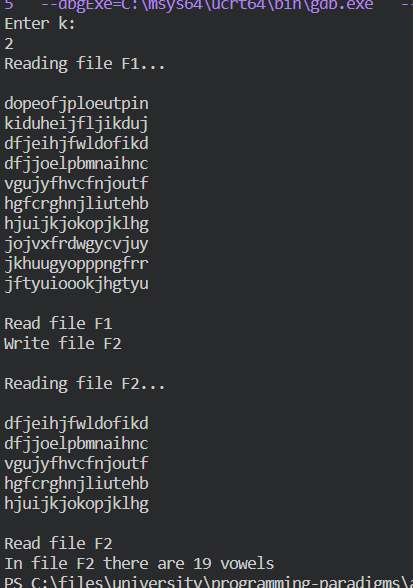
*Малюнок 2. VNS Lab 6 Results*

Завдання №2 VNS Lab 8

* Затрачено: 1 год. 50 хв.
* 

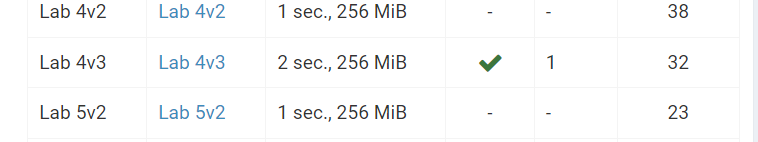
*Малюнок 3. VNS Lab 8 Results*

Завдання №3 VNS Lab 9

* Затрачено: 40 хв.
* 

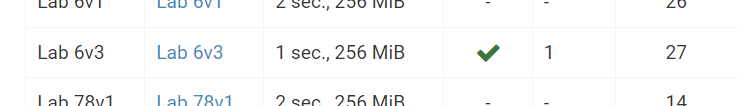
*Малюнок 4. VNS Lab 9 Results*

Завдання №4 Algotester Lab 4

* Затрачено: 1 год. 20 хв.
* 

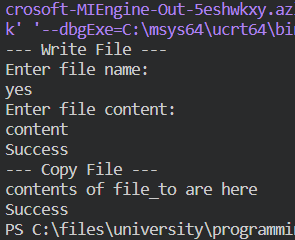
*Малюнок 5. Algotester Lab 4*

Завдання №5 Algotester Lab 6

* Затрачено: 2 год.
* 

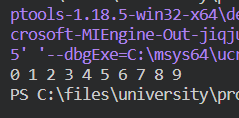
*Малюнок 6. Algotester Lab 6*

Завдання №6 Class Practice Task

* Затрачено: 50 хв.
* 

*Малюнок 7. Class Practice Task*

Завдання №7 Self Practice Task

* Затрачено: 50 хв.
* 

*Малюнок 8. Self Practice Task*

# **Висновки:**

Ознайомився з:

* файлами,
* системами числення,
* бінарними файлами,
* символами і рядковими змінними та текстовими файлами,
* стандартною бібліотекою,
* методами роботи з файлами,
* створенням і використанням бібліотек.

Успішно написав код з використанням нового матеріалу.