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 та Рядкові Змінні. Текстові Файли. Стандартна бібліотека. Створення й використання бібліотек.»

***Виконала:***

студентка групи ШІ-14

Павлюх Вікторія Ігорівна

# **Тема роботи:**

Текстові та бінарні файли, символи та рядкові символами; системи числення та стандартна бібліотека.

# **Мета роботи:**

# Навчитися працювати з файлами(бінарними і текстовими) (зчитувати та записувати файли).Розглянути концепцію символів та рядкових змінних, їх операцій та використання у програмуванні.Освоїти стандартні бібліотеки С++, що включають функції для оптимізації обробки файлів та рядків.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Текстові та бінарні файли.
* Тема №2: Символи та рядкові змінні.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Текстові та бінарні файли.
  + Джерела Інформації
    - Відео 1.<https://youtu.be/Cz4fl-TUjVk>.

2.<https://youtu.be/FeNqHytI0fA?feature=shared>

* + Що опрацьовано:
    - Як записувати інформацію в файли та зчитувати її з них
  + Статус: Ознайомлена
  + Витрачений час: тиждень
* Тема №2: Символи та рядкові змінні.
  + - Стаття 1.<https://acode.com.ua/urok-60-vvedennya-v-std-string/>

2.http:[//www.kievoit.ippo.kubg.edu.ua/kievoit/2016/73\_C++/index.html](http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/73_C++/index.html)

* + Що опрацьовано:
    - Як реалізовані рядкові змінні в С++, як ними користуватися, корисні функції пов’язані зі string.
  + Статус: Ознайомлена
  + Витрачений час: тиждень

# **Виконання роботи:**

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

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

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

## **4. Результати виконання завдань, тестування.**

Завдання №1 **VNS Lab 6 - Task 1-N**

* Варіант завдання – 21
* Деталі завдання: Задано рядок, що складається із символів. Символи поєднуються в слова. Слова одне від одного відокремлюються одним або декількома пробілами. Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів. Знищити всі парні слова у речені.
* Блок-схема: не робила
* Код програми:

*#include* <iostream>

*#include* <vector>

*#include* <string>

using namespace std;

*string* remove\_paired\_words(vector<char>& *charVector*) {

int count=0;

int temp=0;

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

if(*charVector*[i]==' ') {

count++;

if (count % 2 != 0) {

temp = i;

}else if (count % 2 == 0){

*charVector*.erase(*charVector*.begin() + temp, *charVector*.begin() + i );

i = temp + 1;

temp = 0;

}

}

}

*charVector*.erase(*charVector*.begin() + *charVector*.size()-1);

return string(*charVector*.begin(), *charVector*.end());

}

int main() {

*string* str;

cout << "Enter a sentence: ";

*// hello my dear friend how are you*

getline(cin, str);

vector<char> charVector;

charVector.assign(str.begin(), str.end());

charVector.push\_back(' ');

*string* processed\_sentence = remove\_paired\_words(charVector);

cout << "Processed sentence: " << processed\_sentence << "." << endl;

return 0;

}

* Результат:

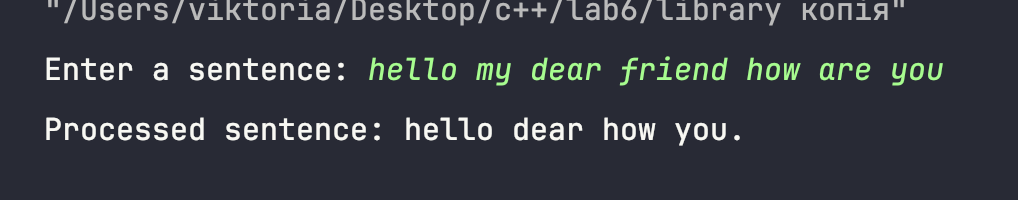


Рис.1.Результат тестування VNS Lab 6 - Task 1-N.

Завдання №2 **VNS Lab 8 - Task 1-N**

* Варіант завдання - 21
* 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: "Автомобіль":

- марка;

- серійний номер;

- реєстраційний номер;

- рік випуску.

Знищити 3 елементи з початку файлу, додати елемент поле елемента із зазначеним реєстраційним номером

* Блок-схема: не робила
* Код програми:

*#include* <iostream>

*#include* <vector>

*#include* <fstream>

*#include* <sstream>

using namespace std;

struct Car {

char brand[20];

char serialNumber[20];

int registrationNumber;

int year;

};

void File(const vector<Car>& *cars*, const *string*& *FileName*)

{

*ofstream* file(*FileName*);

if (!file)

{

cerr << "An error occurred while opening/creating the file" << endl;

return;

}

for (const Car& c : *cars*)

{

file << c.brand << "\t" << c.serialNumber << "\t" << c.registrationNumber << "\t" << c.year << endl;

}

file.close();

}

void DeleteCars(vector<Car>& *cars*, const char\* *FileName*)

{

if (*cars*.size() >= 3)

{

*cars*.erase(*cars*.begin(), *cars*.begin() + 3);

}

File(*cars*, *FileName*); *// Викликаємо функцію File для оновлення файлу*

}

bool AddCar(const char\* *FileName*)

{

Car newcar;

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

cout << "brand: ";

cin >> newcar.brand;

cout << "serialNumber: ";

cin >> newcar.serialNumber;

cout << "registrationNumber: ";

cin >> newcar.registrationNumber;

cout << "year: ";

cin >> newcar.year;

*string* afterCar;

cout << "After which car do you want to insert a new one?\n";

cin >> afterCar;

*ifstream* file1(*FileName*);

if (!file1)

{

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

return false;

}

vector<Car> cars;

*string* str;

while (getline(file1, str))

{

*istringstream* iss(str);

Car c;

iss >> c.brand >> c.serialNumber >> c.registrationNumber >> c.year;

cars.push\_back(c);

}

file1.close();

*ofstream* file2("temp.bin");

if (!file2)

{

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

return false;

}

bool found = false;

for (const Car& c : cars)

{

file2 << c.brand << "\t" << c.serialNumber << "\t" << c.registrationNumber << "\t" << c.year << endl;

if (string(c.brand) == afterCar)

{

file2 << newcar.brand << "\t" << newcar.serialNumber << "\t" << newcar.registrationNumber << "\t" << newcar.year << endl;

found = true;

}

}

if (!found)

{

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

return false;

}

file2.close();

remove(*FileName*);

rename("temp.bin", *FileName*);

return true;

}

void Print(const char\* *FileName*)

{

*ifstream* file(*FileName*);

if (!file)

{

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

return;

}

*string* str;

while (getline(file, str))

{

cout << str << endl;

}

file.close();

}

int main() {

vector <Car> cars = {

{"Volvo", "AK", 2888, 2015},

{"Ford", "AB", 1250, 2000},

{"Mercedes", "CE", 3378, 1999},

{"Zhiguli", "AA", 6499, 1970},

{"Ferrari", "XK", 8822, 1989}

};

File(cars, "file1.bin");

Print("file1.bin");

DeleteCars(cars, "file1.bin");

if (!AddCar("file1.bin")) return 1;

Print("file1.bin");

return 0;

}

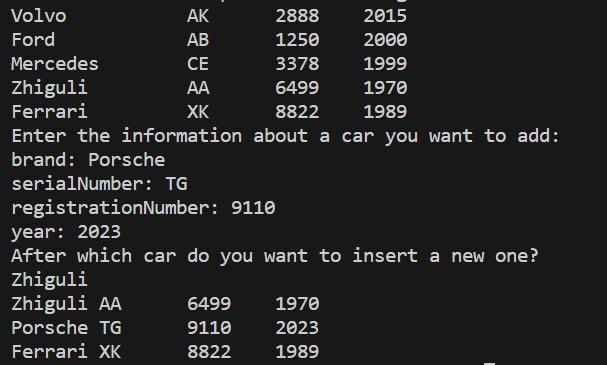
* Результат:
* 

Рис.2. Результат тестування VNS Lab 8 - Task 1-N.

Завдання №3 **VNS Lab 9 - Task 1-N**

* Варіант завдання - 18
* 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, у яких немає однакових слів. Визначити кількість голосних букв у першому рядку файлу F2.
* Блок-схема: не робила
* Код програми:

*#include* <string>

*#include* <vector>

*#include* <fstream>

*#include* <sstream>

using namespace std;

int countVowels(const *string*& *word*) {

int vowelCount = 0;

for (char ch : *word*) {

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

vowelCount++;

}

}

return vowelCount;

}

int main() {

*string* str[] =

{ "Maids table.\n",

"Music.\n",

"Hand dear so we hour to. He we be hastily offence effects he service.\n",

"Sympathize it projection ye insipidity celebrated my pianoforte indulgence.\n",

"Point.\n",

"Elegance exercise as laughing proposal mistaken if.\n",

"We up precaution an it solicitude acceptance invitation.\n",

"Am terminated it excellence invitation projection as.\n",

"Supply.\n",

"Design for are edward regret met lovers.\n"

};

*// Writing to F1.txt*

*ofstream* file1("F1.txt");

if (!file1) {

cerr << "An error occurred " << endl;

return 1;

}

for (int i = 0; i < 10; ++i) {

file1 << str[i] << endl;

}

file1.close();

*// Task 1:*

*ifstream* file1\_1("F1.txt");

*ofstream* file2("F2.txt");

if (!file1\_1 || !file2) {

cerr << "An error occurred" << endl;

return 1;

}

*string* line1;

while (getline(file1\_1, line1)) { *//Зчитується кожен рядок з файлу "F1.txt"*

*// і зберігається в рядку line1.*

*istringstream* iss(line1); *//розбиваємо line1 на слова.*

*string* word; *//буде використовуватися для зберігання кожного окремого*

*// слова під час розбиття рядка.*

int wordCount = 0;

while (iss >> word) { *//iss (відокремлений рядок line1) розбивається на*

*// окремі слова, які записуються у word*

wordCount++;

}

if (wordCount > 2) {

file2 << line1 << endl;

}

}

file2.close();

file1\_1.close();

*// Task 2:*

*ifstream* file2\_2("F2.txt");

if (!file2\_2) {

cerr << "An error occurred" << endl;

return 1;

}

int maxVowelCount = 0;

int wordNumber = 0;

*string* line2;

while (getline(file2\_2, line2)) {

*istringstream* iss(line2);

*string* word;

int currentWordNumber = 0;

while (iss >> word) {

currentWordNumber++;

int vowelCount = countVowels(word);

if (vowelCount > maxVowelCount) {

maxVowelCount = vowelCount;

wordNumber = currentWordNumber;

}

}

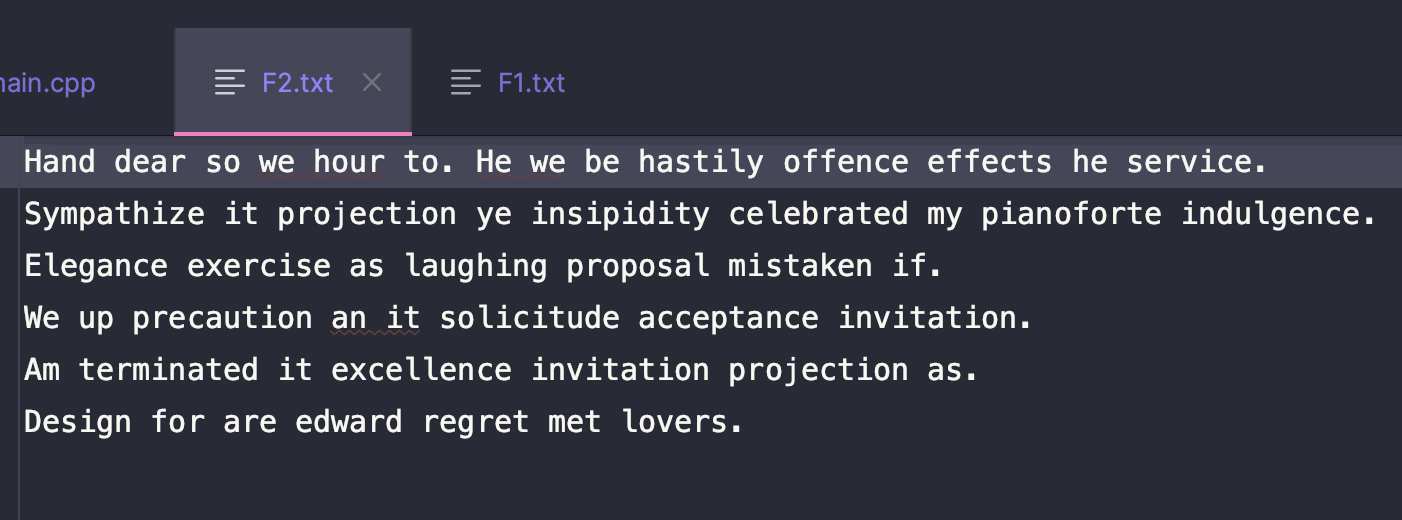
}

file2\_2.close();

cout << "The word with the most vowels is at position " << wordNumber << " with " << maxVowelCount << " vowels." << endl;

return 0;

}

* Результат:

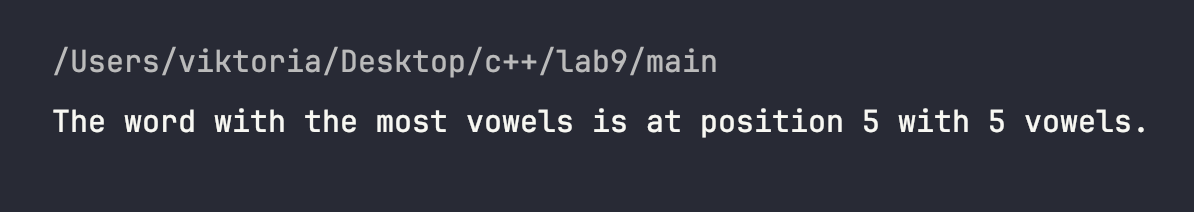


Рис.3-4. Результат тестування VNS Lab 9 - Task 1-N.

Завдання №4 **Algotester Lab 4v2 (варіант 1)**

* Варіант завдання – 2
* Деталі завдання: Вам дано масив a з N цілих чисел. Спочатку видаліть масиву a усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4]. Після цього оберніть посортовану версію масиву a на K, тобто при K = 3 масив [1, 2, 3, 4, 5, 6, 7] перетвориться на [4, 5, 6, 7, 1, 2, 3]. Виведіть результат
* Блок-схема: не робила
* Код програми:

*#include* <vector>

*#include* <algorithm>

using namespace std;

int main() {

int N, K;

cin >> N;

int temp = N;

cin >> K;

vector<int> a(N);

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

cin >> a[i];

}

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

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

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

a.erase(a.begin() + j);

N--;

j--;

}

}

}

if(K>a.size()) {

K = K % N;

}

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

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

cout<<a.size();

cout<<endl;

for (int i : a) {

cout << i << " ";

}

cout << endl;

return 0;

}

* Результат:

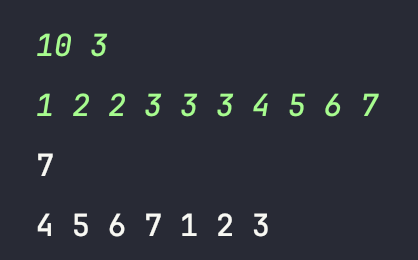


Рис.5. Результат тестування Algotester Lab 4v2 (варіант 1).

Завдання №5 **Algotester Lab 4v2 (варіант 2)**

* Варіант завдання – 2
* Деталі завдання: Вам дано масив a з N цілих чисел. Спочатку видаліть масиву a усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4]. Після цього оберніть посортовану версію масиву a на K, тобто при K = 3 масив [1, 2, 3, 4, 5, 6, 7] перетвориться на [4, 5, 6, 7, 1, 2, 3]. Виведіть результат
* Блок-схема:

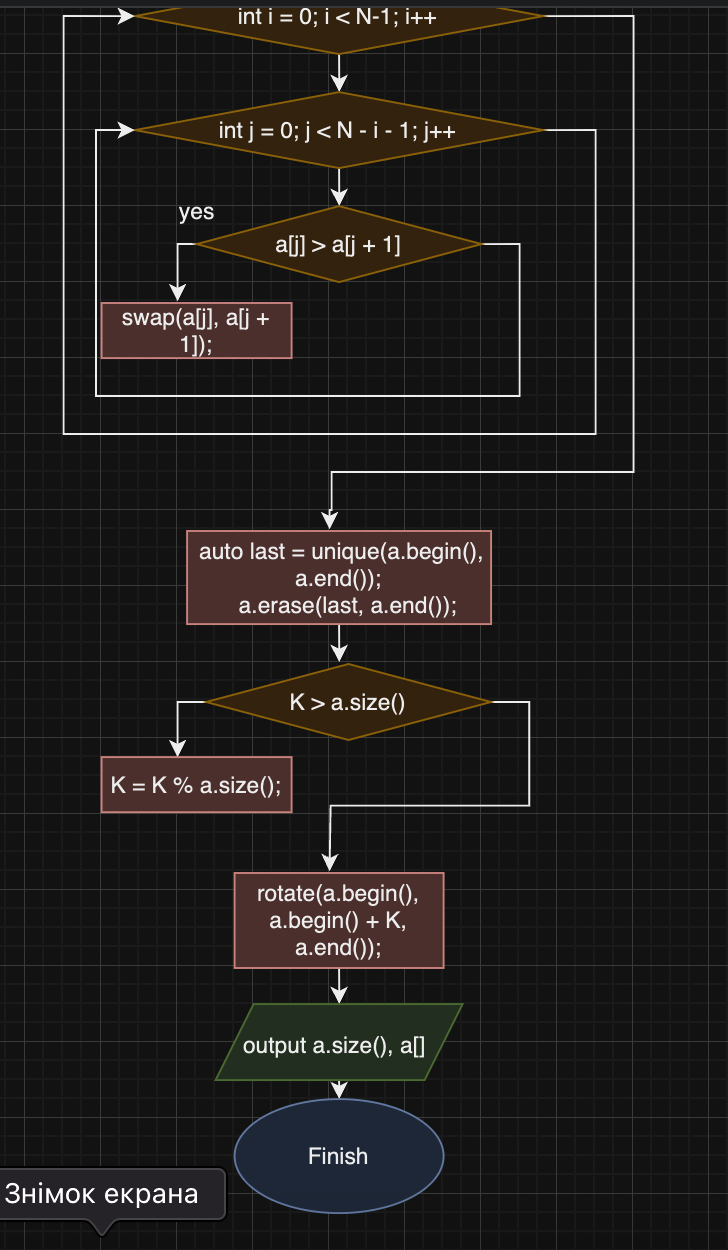
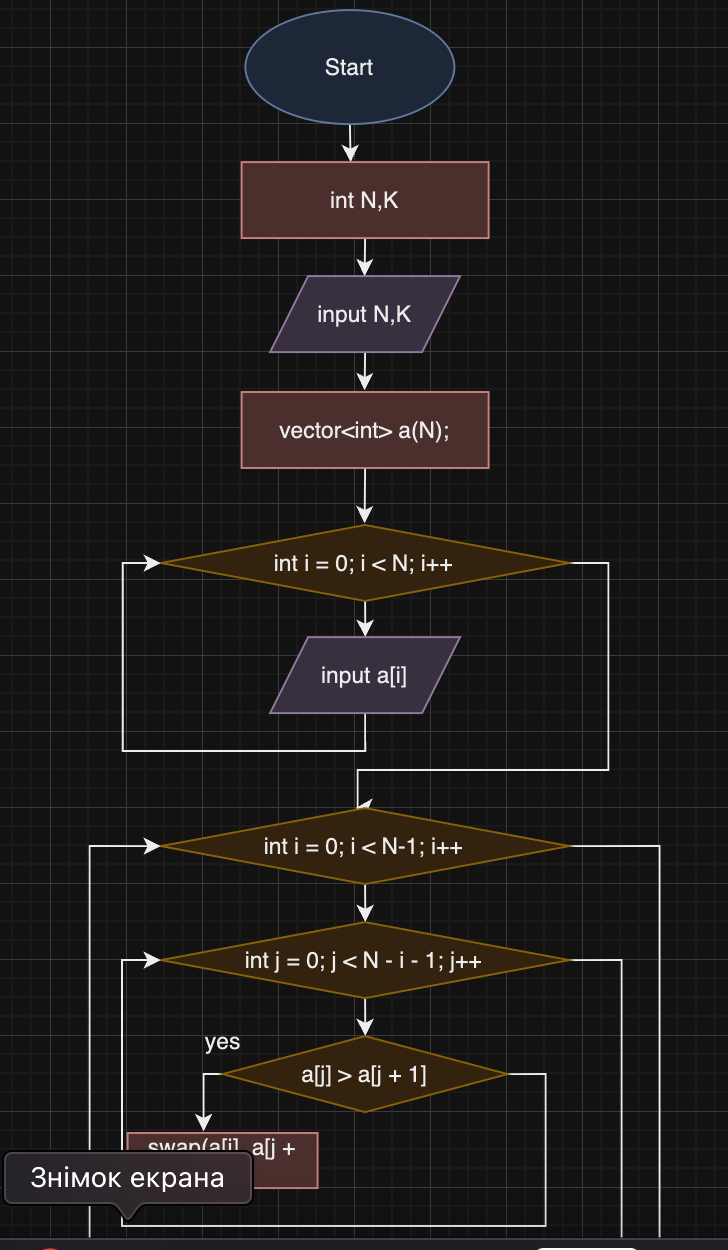


Рис.6-7.Блок-схема до Algotester Lab 4v2 (варіант 2).

* Код програми:

*#include* <iostream>

*#include* <vector>

*#include* <algorithm>

using namespace std;

int main() {

int N, K;

cin >> N >> K;

vector<int> a(N);

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

cin >> a[i];

}

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

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

if (a[j] > a[j + 1]) {

swap(a[j], a[j + 1]);

}

}

}

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

a.erase(last, a.end());

if (K > a.size()) {

K = K % a.size();

}

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

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

for (int i : a) {

cout << i << " ";

}

cout << endl;

return 0;

}

* Результат: такий же, як і в минулій.

Завдання №6 **Algotester Lab 6v2**

* Варіант завдання – 2
* Деталі завдання: У вас є шахова дошка розміром 8×8 та дуже багато фігур. Кожна клітинка може мати таке значення:
* Пуста клітинка O
* Пішак P
* Тура R
* Кінь N
* Слон B
* Король K
* Королева Q

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1). Далі йдуть Q запитів з координатами клітинки {x,y}. На кожен запит ви маєте вивести стрічку si - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз).У випадку, якщо на клітинці стоїть якась фігура - виведіть символ X.У випадку, якщо клітинку не атакують - виведіть O.

Наявніть фігури у певній клітинці не блокує атаку для іншої фігури. Тобто якщо між турою та клітинкою стоїть інша фігура - вважається що тура атакує цю клітинку.

* Блок-схема: не робила
* Код програми:

*#include* <iostream>

*#include* <vector>

*#include* <algorithm>

using namespace std;

int main() {

int q, i, j;

char row[8][8];

vector <vector<*string*> > result(8,vector<*string*>(8, ""));

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

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

cin>>row[i][j];

}

}

cin>>q;

int \*k = new int [q];

int \*t = new int [q];

for (int m = 0; m < q ; m++) {

cin >> k[m] >> t[m];

k[m]--;

t[m]--;

if (row[k[m]][t[m]] != 'O') {

result[k[m]][t[m]] += "X";

}

}

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

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

if (row[i][j] == 'K') {

for (int m = -1; m <= 1; ++m) {

for (int n = -1; n <= 1; ++n) {

int newK = i + m;

int newT = j + n;

if (m != 0 || n != 0) {

if (newK >= 0 && newK < 8 && newT >= 0 && newT < 8 && result[newK][newT] != "X" && result[newK][newT] != "K") {

result[newK][newT] += "K";

}

}

}

}

} else if (row[i][j] == 'R') {

for (int m = 0; m < 8; m++) {

if (m != i && result[m][j] != "X" && result[m][j] != "R") {

result[m][j] += "R";

}

}

for (int m = 0; m < 8; m++) {

if (m != j && result[i][m] != "X" && result[i][m] != "R") {

result[i][m] += "R";

}

}

} else if (row[i][j] == 'B') {

for ( int m = i-1, n = j+1; m >= 0 && n < 8; m--, n++) {

if (result[m][n] != "X" && result[m][n] != "B") {

result[m][n] += "B";

}

}

for ( int m = i+1, n = j-1; m < 8 && n >= 0 ; m++, n--) {

if (result[m][n] != "X" && result[m][n] != "B") {

result[m][n] += "B";

}

}

for ( int m = i-1, n = j-1; m >= 0 && n >= 0; m--, n--) {

if (result[m][n] != "X" && result[m][n] != "B") {

result[m][n] += "B";

}

}

for ( int m = i+1, n = j+1; m < 8 && n < 8; m++, n++) {

if (result[m][n] != "X" && result[m][n] != "B") {

result[m][n] += "B";

}

}

} else if (row[i][j] == 'N') {

int x[8] = {-1, -2, -2, -1, 1, 2, 2, 1};

int y[8] = {-2, -1, 1, 2, -2, -1, 1, 2};

for (int m = 0; m < 8; m++) {

int newX = i + x[m];

int newY = j + y[m];

if (newX >= 0 && newX < 8 && newY >= 0 && newY < 8 && result[newX][newY] != "X" && result[newX][newY] != "N") {

result[newX][newY] += "N";

}

}

} else if (row[i][j] == 'Q') {

for ( int m = i-1, n = j+1; m >= 0 && n < 8; m--, n++) {

if (result[m][n] != "X" && result[m][n] != "Q") {

result[m][n] += "Q";

}

}

for ( int m = i+1, n = j-1; m < 8 && n >= 0 ; m++, n--) {

if (result[m][n] != "X" && result[m][n] != "Q") {

result[m][n] += "Q";

}

}

for ( int m = i-1, n = j-1; m >= 0 && n >= 0; m--, n--) {

if (result[m][n] != "X" && result[m][n] != "Q") {

result[m][n] += "Q";

}

}

for ( int m = i+1, n = j+1; m < 8 && n < 8; m++, n++) {

if (result[m][n] != "X" && result[m][n] != "Q") {

result[m][n] += "Q";

}

}

for (int m = 0; m < 8; m++) {

if (m != i && result[m][j] != "X" && result[m][j] != "Q") {

result[m][j] += "Q";

}

}

for (int m = 0; m < 8; m++) {

if (m != j && result[i][m] != "X" && result[i][m] != "Q") {

result[i][m] += "Q";

}

}

} else if (row[i][j] == 'P') {

if (i + 1 < 8 && j - 1 >= 0 && result[i+1][j-1] != "X" && result[i+1][j-1] != "P") {

result[i+1][j-1] += "P";

}

if (i + 1 < 8 && j + 1 < 8 && result[i+1][j+1] != "X" && result[i+1][j+1] != "P") {

result[i+1][j+1] += "P";

}

}

}

}

for(int m=0; m<8; m++) {

for(int n=0; n<8; n++) {

if (result[m][n]==""){

result[m][n]="O";

}

sort(result[m][n].begin(), result[m][n].end());

}

}

for(int m=0; m<q; m++){

cout<<result[k[m]][t[m]]<<endl;

}

return 0;

}

* Результат:

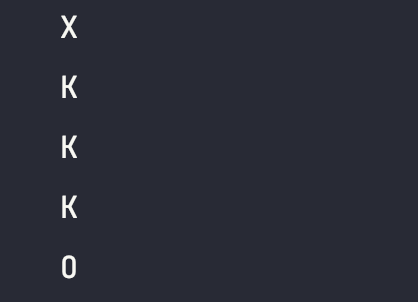


Рис.8. Результат тестування Algotester Lab 6v2.

Завдання №7 **Class Practice Work**

* Деталі завдання: Реалізувати функцію створення файла і запису в нього даних: запис текстової стрічки у файл із заданим ім’ям; копіювання вмісту файла у інший файл.
* Блок-схема: не робила
* Код програми:

*#include* <iostream>

*#include* <fstream>

*#include* <sstream>

*#include* <cctype>

using namespace std;

enum FileOpResult {*Success*, *Failure*};

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

*ofstream* file1(*name*);

if (!file1) return *Failure*;

file1 << *content*;

file1.close();

return *Success*;

}

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

*ifstream* file1(*file\_from*);

*ofstream* file2(*file\_to*);

if (!file1)

{

cerr << "File " << *file\_from* << " does not exist" << endl;

file1.close();

return *Failure*;

}

if (!file2)

{

cerr << "An error occured while writing into the file " << *file\_to* << endl;

file2.close();

return *Failure*;

}

*string* str;

while (getline(file1, str))

{

file2 << str << endl;

}

file1.close(), file2.close();

return *Success*;

}

int main() {

char name[] = "F1.txt";

char content[100];

cout << "enter the content, which you want to write to file" << endl;

cin.getline(content, sizeof(content));

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

if(res == *Success*){

cout << "successful!" << endl;

} else{

cout << "failed!" << endl;

}

return 0;

}

* Результат:

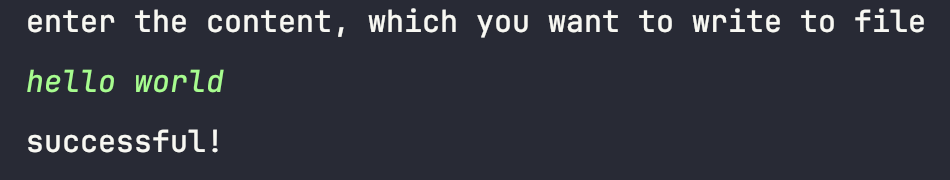


Рис.9.Результат тестування Class Practice Work.

Завдання №8 **Self-Practice Work**

* Варіант завдання: **The Algo Battles 2023 - Етап 7 - А**
* Деталі завдання: Петрик любить грати в ігри, де світ складається з кубів. Сьогодні він вирішив написати свій варіант такої гри. Петрик хоче, щоб у цій грі можна було програмувати нескладні механізми на основі електричного сигналу. Вважається, що сигнал має певну додатну силу, яка позначається цілим числом. Також, сигнал може бути відсутнім, що позначається як сигнал силою 0. Петрик додав в гру три блоки, які по-різному обробляють електричний сигнал:
  + Кабель отримує сигнал силою k і видає сигнал силою max(k−1,0).
  + Повторювач підсилює сигнал до рівня 15 при ненульовому вхідному сигналі, і не видає сигнал при його відсутності. Формально, нехай сила вхідного сигналу k. Тоді повторювач видає сигнал силою 15, якщо k≥1, і сигнал силою 0, якщо k=0.
  + Змінювач не видає сигнал при ненульовому вхідному сигналі та видає сигнал силою 15 при відсутності вхідного сигналу. Формально, нехай сила вхідного сигналу k. Тоді змінювач видає сигнал силою 0, якщо k≥1, і сигнал силою 15, якщо k=0.

Петрик задає вам електричну лінію, в якій кабелі, повторювачі і змінювачі позначені літерами k, p, z відповідно. На перший елемент лінії подано сигнал силою 15. Допоможіть Петрику знайти вихідний сигнал останнього блоку в лінії.

* Блок-схема: не робила
* Код програми:

*#include* <iostream>

*#include* <string>

using namespace std;

int main() {

int input\_signal=15;

int n;

cin>>n;

char \*s=new char[n];

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

cin>>s[i];

}

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

if(s[i]=='k'){

if(input\_signal!=0){

input\_signal-=1;

}

}else if(s[i]=='p'){

if(input\_signal!=0){

input\_signal=15;

}

}else if(s[i]=='z'){

if(input\_signal!=0){

input\_signal=0;

}else if(input\_signal==0) {

input\_signal=15;

}

}

}

cout<<input\_signal;

return 0;

}

* Результат:

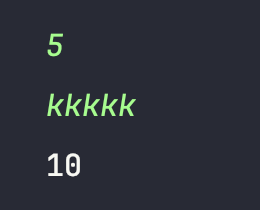


Рис.10.Результат тестування Self-Practice Work.

# **Висновки:**

Навчилася працювати з текстовими та бінарними файлами, а саме зчитувати та записувати інформацію з/в них.