

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 5

На тему: «Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли.
Стандартна бібліотека та деталі/методи роботи з файлами. Створення й
використання бібліотек.»

з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи № 6

ВНС Лабораторної Роботи № 8

ВНС Лабораторної Роботи № 9

Алготестер Лабораторної Роботи №4

Алготестер Лабораторної Роботи №6

Практичних Робіт до блоку №5

Виконав:

Студент групи ШІ-11

Саух Богдан

Львів 2024

Тема роботи: Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

Мета роботи: Навчитися записувати і зчитувати інформацію з файлу стилями мов С та С++. Базово розібратися що таке бібліотека і де \\ використовують.

Теоретичні відомості:

1. Лекції, практичні
2. aCode
3. ChatGPT

Виконання роботи

Завдання 1: Class Practice Work

Задача 1

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 файлу.

Задача 2

Реалізувати функцію створення файла і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };
```

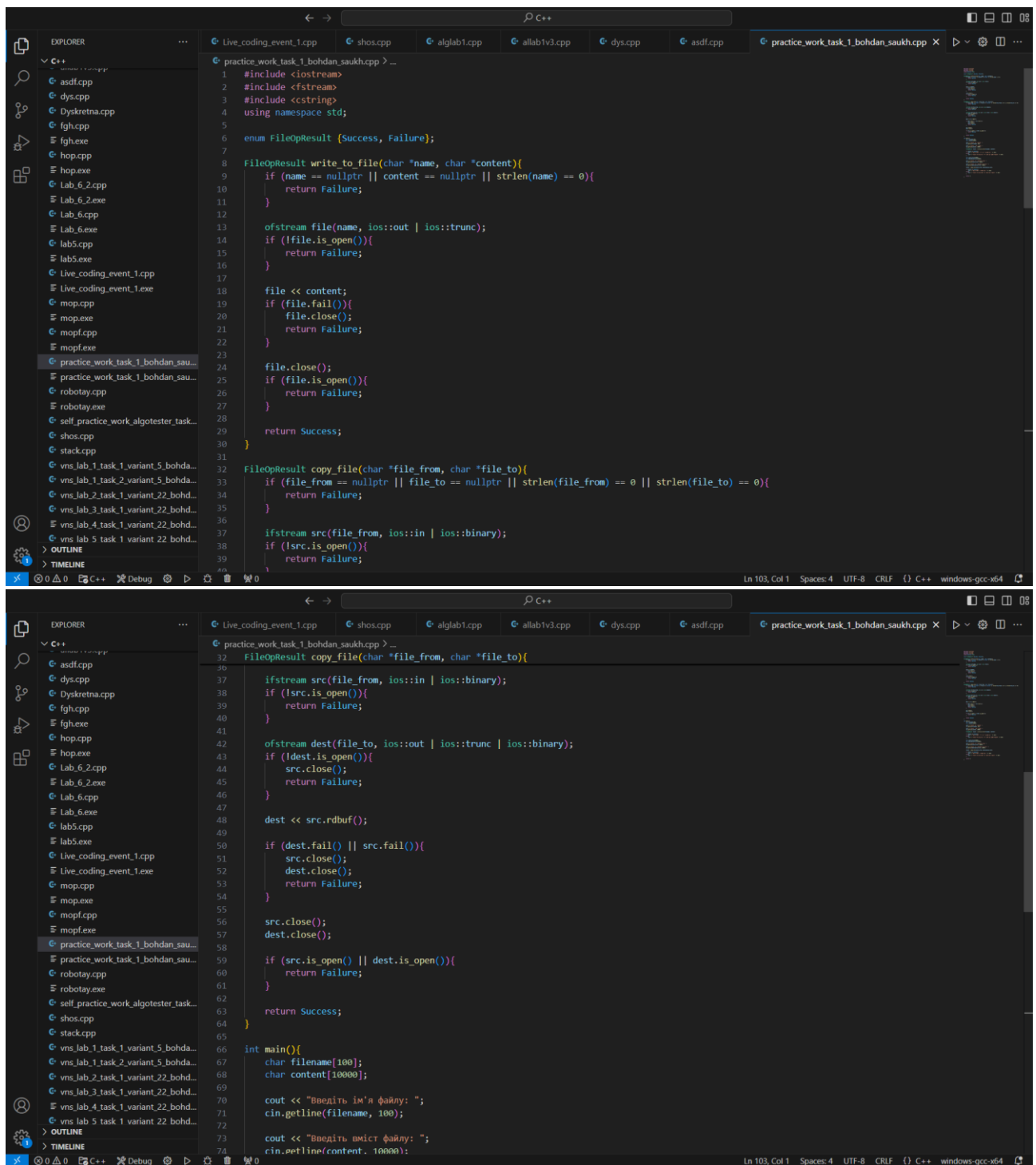
```
FileOpResult copy_file(char *file_from, char *file_to);
```

Умови задачі:

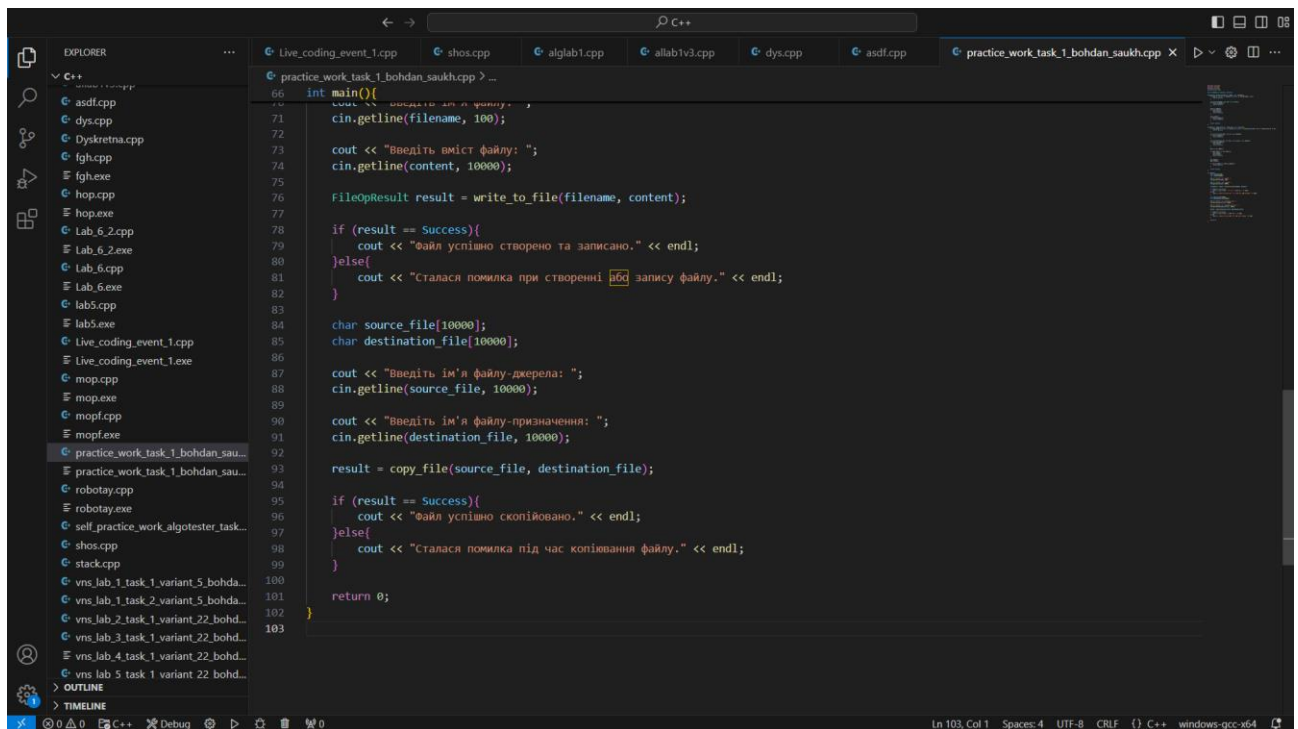
- копіювати вміст файлу з ім'ям file_from у файл з ім'ям file_to; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів
- file_from, file_to – можуть бути повним або відносним шляхом

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

Код:

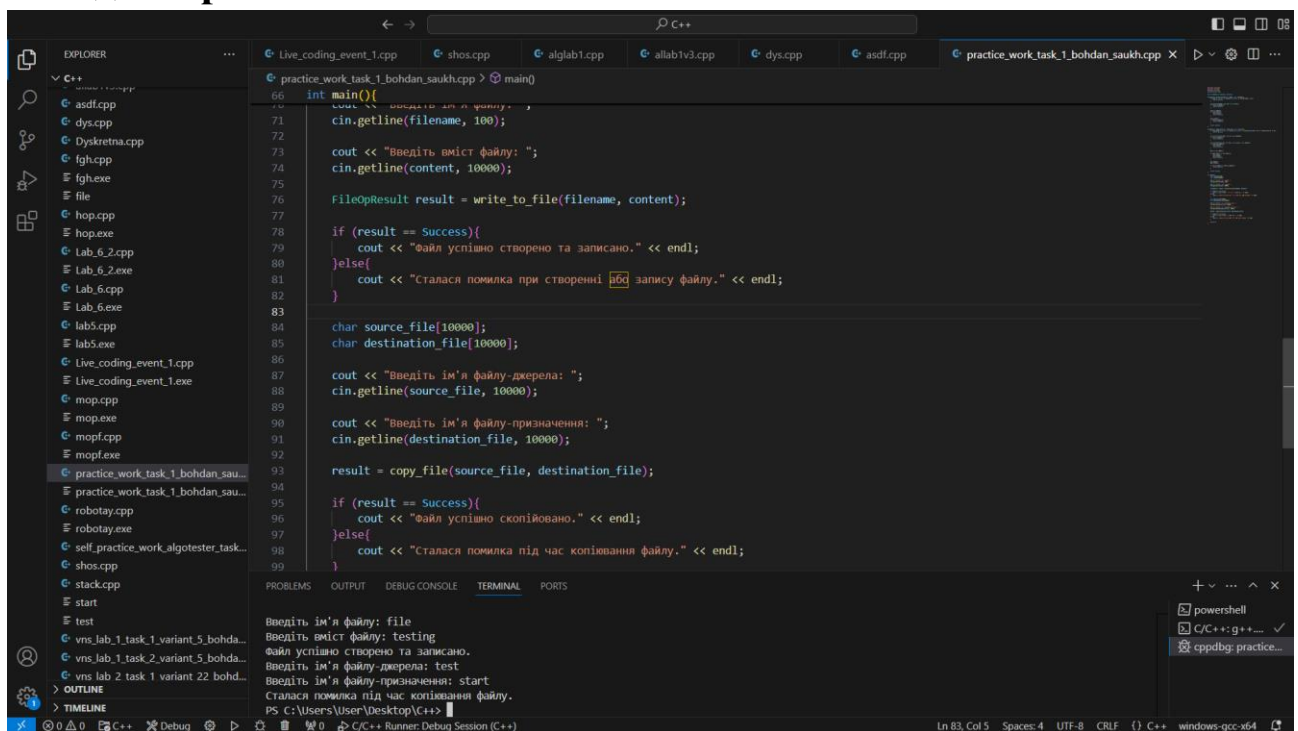


```
1 #include <iostream>
2 #include <fstream>
3 #include <string>
4 using namespace std;
5
6 enum FileOpResult (Success, Failure);
7
8 FileOpResult write_to_file(char *name, char *content){
9     if (name == nullptr || content == nullptr || strlen(name) == 0){
10         return Failure;
11     }
12
13     ofstream file(name, ios::out | ios::trunc);
14     if (!file.is_open()){
15         return Failure;
16     }
17
18     file << content;
19     if (file.fail()){
20         file.close();
21         return Failure;
22     }
23
24     file.close();
25     if (file.is_open()){
26         return Failure;
27     }
28
29     return Success;
30 }
31
32 FileOpResult copy_file(char *file_from, char *file_to){
33     if (file_from == nullptr || file_to == nullptr || strlen(file_from) == 0 || strlen(file_to) == 0){
34         return Failure;
35     }
36
37     ifstream src(file_from, ios::in | ios::binary);
38     if (!src.is_open()){
39         return Failure;
40     }
41
42     ofstream dest(file_to, ios::out | ios::trunc | ios::binary);
43     if (!dest.is_open()){
44         src.close();
45         return Failure;
46     }
47
48     dest << src.rdbuf();
49
50     if (dest.fail() || src.fail()){
51         src.close();
52         dest.close();
53         return Failure;
54     }
55
56     src.close();
57     dest.close();
58
59     if (src.is_open() || dest.is_open()){
60         return Failure;
61     }
62
63     return Success;
64 }
65
66 int main(){
67     char filename[100];
68     char content[10000];
69
70     cout << "Введіть ім'я файлу: ";
71     cin.getline(filename, 100);
72
73     cout << "Введіть зміст файлу: ";
74     cin.getline(content, 10000);
```



```
66 int main(){
67     cout << "Введіть ім'я файлу: ";
68     cin.getline(filename, 100);
69
70     cout << "Введіть вміст файлу: ";
71     cin.getline(content, 10000);
72
73     FileOpResult result = write_to_file(filename, content);
74
75     if (result == Success){
76         cout << "Файл успішно створено та записано." << endl;
77     }else{
78         cout << "Сталася помилка при створенні або запису файлу." << endl;
79     }
80
81     char source_file[10000];
82     char destination_file[10000];
83
84     cout << "Введіть ім'я файлу-джерела: ";
85     cin.getline(source_file, 10000);
86
87     cout << "Введіть ім'я файлу-призначення: ";
88     cin.getline(destination_file, 10000);
89
90     result = copy_file(source_file, destination_file);
91
92     if (result == Success){
93         cout << "Файл успішно скопійовано." << endl;
94     }else{
95         cout << "Сталася помилка під час копіювання файлу." << endl;
96     }
97
98     return 0;
99 }
```

Вивід в терміналі:



```
Введіть ім'я файлу: file
Введіть вміст файлу: testing
Файл успішно створено та записано.
Введіть ім'я файлу-джерела: test
Введіть ім'я файлу-призначення: start
Сталася помилка під час копіювання файлу.
PS C:\Users\User\Desktop\C++>
```

Час виконання ~ 2 години

Завдання 2: VNS Lab 6 - Task 1-22

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.

Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку рядка у відповідності зі своїм варіантом.

Для рядка знайти кількість слів, що починаються на букву «а».

Код:

Вивід в терміналі:

```
PS C:\Users\User\Desktop\C++> & "c:\Users\User\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe" "-stdin-Microsoft-MiEngine-In-n34gtv30.shp" "--stdout-Microsoft-MiEngine-Out-xpkpro8.bsp" "--stderr-Microsoft-MiEngine-Error-wthjffjon.ztm" "--pid-Microsoft-MiEngine-Pid-hshpd4qu.gog" "--dbgExe=c:\msys64\usr\bin\gdb.exe" "--interpreter=mi" asdf fgh awdcc Afgre ertyuadf Кількість слів, що починаються на букву 'A': 3 PS C:\Users\User\Desktop\C++>
```

Час виконання ~ 10 хв

Завдання 3: VNS Lab 8 - Task 1-22

Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вміст, 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 "Власник автомобіля":

- прізвище, ім'я, по батькові;
- номер автомобіля;
- номер техпаспорта;
- відділення реєстрації ДАІ.

Знищити елемент із заданим номером, додати 2 елементи перед елементом із заданим прізвищем.

Код:

```
1 #include <iostream>
2 #include <fstream>
3 #include <vector>
4 #include <cstring>
5
6 using namespace std;
7
8 struct Owner{
9     char second_name[50];
10    char name[50];
11    char fathers_name[50];
12    char car_number[50];
13    char tech_passport[50];
14    char dai[50];
15 };
16
17 void create_binary_file(const char* filename, const vector<Owner>& owners){
18     ofstream file(filename, ios::binary);
19     for (const auto& owner : owners) {
20         file.write(reinterpret_cast<const char*>(&owner), sizeof(Owner));
21     }
22     file.close();
23 }
24
25 void print_binary_file(const char* filename){
26     ifstream file(filename, ios::binary);
27     Owner owner;
28     while (file.read(reinterpret_cast<char*>(&owner), sizeof(Owner))){
29         cout << owner.second_name << " " << owner.name << " " << owner.fathers_name << endl;
30         cout << owner.car_number << ", " << owner.tech_passport << ", " << owner.dai << endl << endl;
31     }
32     file.close();
33 }
34
35 void delete_by_car_number(const char* filename, const char* car_number){
36     ifstream file(filename, ios::binary);
37     vector<Owner> owners;
38     Owner owner;
39     while (file.read(reinterpret_cast<char*>(&owner), sizeof(Owner))){
40         if (strcmp(owner.car_number, car_number) != 0) {
41             owners.push_back(owner);
42         }
43     }
44     file.close();
45     create_binary_file(filename, owners);
46 }
```

```
void delete_by_car_number(const char* filename, const char* car_number){
    while (file.read(reinterpret_cast<char*>(&owner), sizeof(owner))) {
        if (strcmp(owner.car_number, car_number) != 0) {
            owners.push_back(owner);
        }
    }
    file.close();

    ofstream outfile(filename, ios::binary | ios::trunc);
    for (const auto& o : owners) {
        outfile.write(reinterpret_cast<const char*>(&o), sizeof(owner));
    }
    outfile.close();
}

void add_before_second_name(const char* filename, const char* target_second_name, const vector<Owner>& new_owners){
    ifstream file(filename, ios::binary);
    vector<Owner> owners;
    Owner owner;

    while (file.read(reinterpret_cast<char*>(&owner), sizeof(owner))) {
        if (strcmp(owner.second_name, target_second_name) == 0) {
            owners.insert(owners.end(), new_owners.begin(), new_owners.end());
        }
        owners.push_back(owner);
    }
    file.close();

    ofstream outfile(filename, ios::binary | ios::trunc);
    for (const auto& o : owners) {
        outfile.write(reinterpret_cast<const char*>(&o), sizeof(owner));
    }
    outfile.close();
}

int main() {
    const char* filename = "owners.dat";
```

```
int main() {
    const char* filename = "owners.dat";

    vector<Owner> owners = {
        {"Smith", "John", "James", "AA1234BB", "TP001", "New York"},
        {"Johnson", "Michael", "Edward", "BB5678CC", "TP002", "Los Angeles"},
        {"Brown", "Christopher", "Daniel", "CC9012DD", "TP003", "Chicago"}
    };

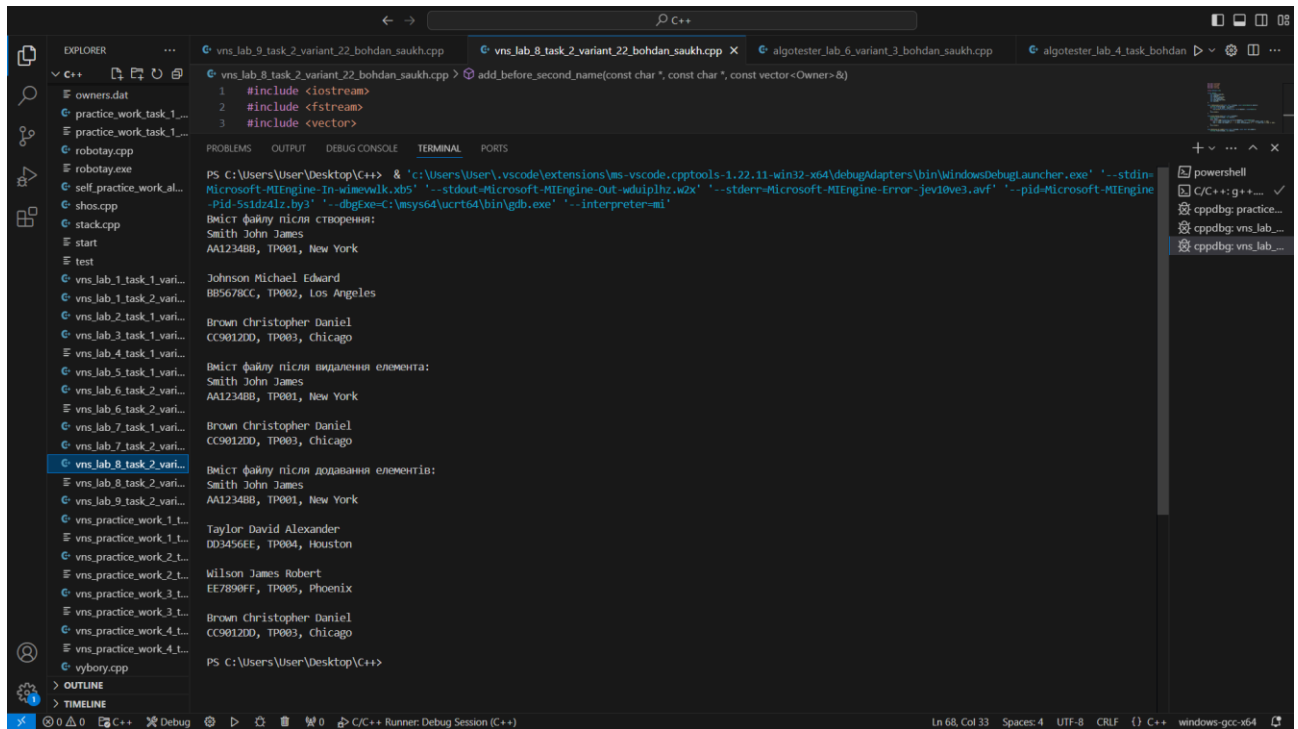
    create_binary_file(filename, owners);
    cout << "Вміст файлу після створення:" << endl;
    print_binary_file(filename);

    delete_by_car_number(filename, "BB5678CC");
    cout << "Вміст файлу після видалення елемента:" << endl;
    print_binary_file(filename);

    vector<Owner> new_owners = {
        {"Taylor", "David", "Alexander", "DD3456EE", "TP004", "Houston"},
        {"Wilson", "James", "Robert", "EE7890FF", "TP005", "Phoenix"}
    };
    add_before_second_name(filename, "Brown", new_owners);
    cout << "Вміст файлу після додавання елементів:" << endl;
    print_binary_file(filename);

    return 0;
}
```

Вивід в терміналі:



Час виконання ~ 3 години

Завдання 3: VNS Lab 9 - Task 1-22

Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію

Виконати завдання.

1) Скопіювати з файлу F1 у файл F2 всі рядки, у яких міститься тільки одне слово.

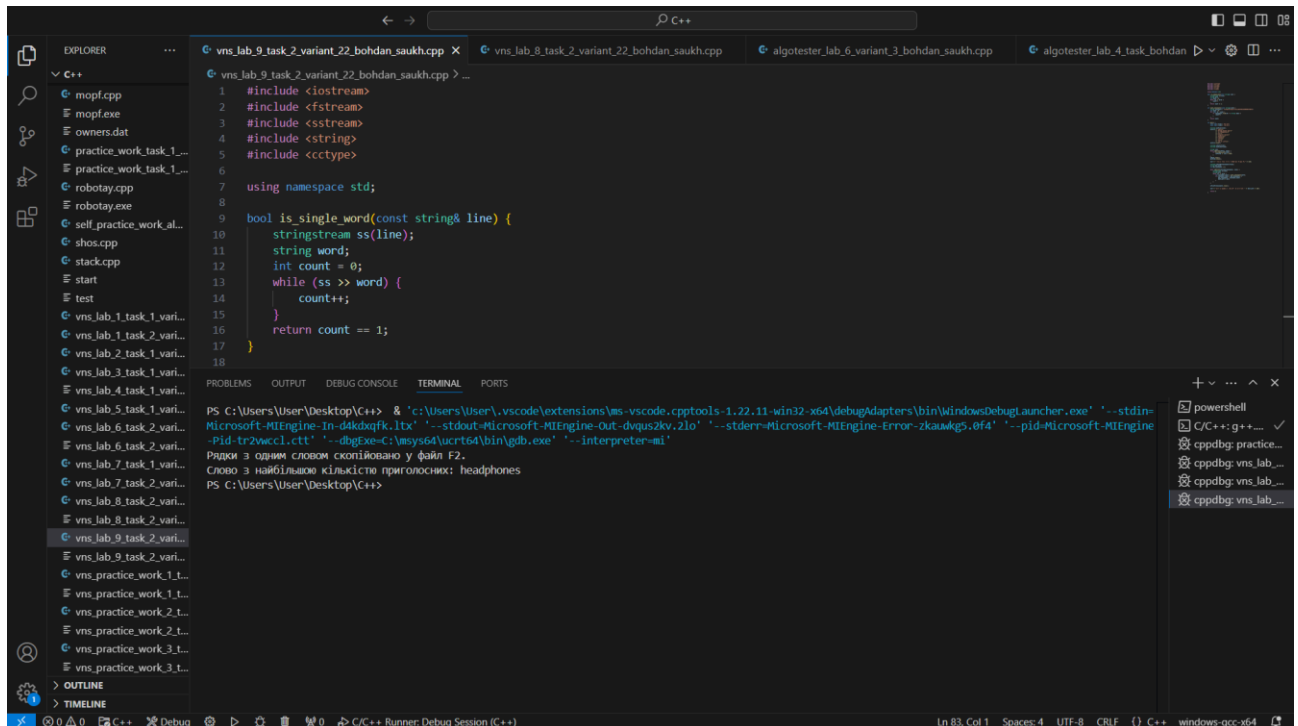
2) Визначити номер слова, у якому найбільше приголосних букв.

Код:


```
1 #include <iostream>
2 #include <fstream>
3 #include <sstream>
4 #include <string>
5 #include <cctype>
6
7 using namespace std;
8
9 bool is_single_word(const string& line) {
10     stringstream ss(line);
11     string word;
12     int count = 0;
13     while (ss >> word) {
14         count++;
15     }
16     return count == 1;
17 }
18
19 int count_consonants(const string& word) {
20     string consonants = "bcdfghijklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ";
21     int count = 0;
22     for (char ch : word) {
23         if (consonants.find(ch) != string::npos) {
24             count++;
25         }
26     }
27     return count;
28 }
29
30 int main() {
31     const char* file1 = "F1.txt";
32     const char* file2 = "F2.txt";
33
34     ofstream outFile(file1);
35     outFile << "Apple\n"
36             << "Orange banana pear\n"
37             << "My headphones\n"
38             << "Car\n"
39             << "Computer mouse\n"
```

```
40     outFile.close();
41
42     ifstream inFile(file1);
43     ofstream outFile2(file2);
44
45     string line;
46     while (getline(inFile, line)) {
47         if (is_single_word(line)) {
48             outFile2 << line << endl;
49         }
50     }
51     inFile.close();
52     outFile2.close();
53
54     cout << "Рядки з одним словом скопійовано в файл F2." << endl;
55
56     ifstream inFileForConsonants(file1);
57     string best_word;
58     int max_consonants = 0;
59
60     while (getline(inFileForConsonants, line)) {
61         stringstream ss(line);
62         string word;
63         while (ss >> word) {
64             int consonant_count = count_consonants(word);
65             if (consonant_count > max_consonants) {
66                 max_consonants = consonant_count;
67                 best_word = word;
68             }
69         }
70     }
71     inFileForConsonants.close();
72
73     cout << "Слово з найбільшою кількістю приголосних: " << best_word << endl;
74
75     return 0;
76 }
77
78
79
80
81
82 }
```

Вивід в терміналі:



```
1 #include <iostream>
2 #include <cstream>
3 #include <cstring>
4 #include <string>
5 #include <cctype>
6
7 using namespace std;
8
9 bool is_single_word(const string& line) {
10     stringstream ss(line);
11     string word;
12     int count = 0;
13     while (ss >> word) {
14         count++;
15     }
16     return count == 1;
17 }
18
```

```
PS C:\Users\User\Desktop\C++> & 'c:\Users\User\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '-.stdin=Microsoft-MIEngine-In-d4kdqfkltx' '-.stdout=Microsoft-MIEngine-Out-dvqus2kv.21o' '-.stderr=Microsoft-MIEngine-Error-zkauwkg5.0f4' '-.pid=Microsoft-MIEngine-Pid-tr2wcc1.ctt' '-.dbgExe=c:\msys64\ucrt64\bin\gdb.exe' '-.interpreter=mi'
Рядки з одним словом скопійовано у файл F2.
Слово з найбільшою кількістю приголосних: headphones
PS C:\Users\User\Desktop\C++>
```

Час виконання ~ 1.5 години

Завдання 4: Algotester Lab 4

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 результируючий масив.

Код з використанням STL:



About

Statistics

Help

Analysis

Events

BOHDANSAUKH

My Solutions

My teams

ARCHIVE

Problems

Rank

Solution Queue

Contact Us

College

Sponsors

Donate

english

Богдан Саук

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4
5 using namespace std;
6
7 int main(){
8     int n;
9     cin >> n;
10    int mas[n];
11    vector<int> mas_0;
12    vector<int> mas_1;
13    vector<int> mas_2;
14
15    for (int i = 0; i < n; i++){
16        cin >> mas[i];
17        if (mas[i] % 3 == 0){
18            mas_0.push_back(mas[i]);
19        }
20        if (mas[i] % 3 == 1){
21            mas_1.push_back(mas[i]);
22        }
23        if (mas[i] % 3 == 2){
24            mas_2.push_back(mas[i]);
25        }
26    }
27
28    sort(mas_2.begin(), mas_2.end());
29    auto it = unique(mas_2.begin(), mas_2.end());
30    mas_2.erase(it, mas_2.end());
31
32    sort(mas_1.begin(), mas_1.end(), greater<int>());
33    auto it_1 = unique(mas_1.begin(), mas_1.end());
34    mas_1.erase(it_1, mas_1.end());
35
36    sort(mas_0.begin(), mas_0.end());
37    auto it_0 = unique(mas_0.begin(), mas_0.end());
38    mas_0.erase(it_0, mas_0.end());
39
40    cout << mas_0.size() + mas_1.size() + mas_2.size() << endl;
41
42    for (int i = 0; i < mas_0.size(); i++){
43        cout << mas_0[i] << " ";
44    }
45    for (int i = 0; i < mas_1.size(); i++){
46        cout << mas_1[i] << " ";
47    }
48    for (int i = 0; i < mas_2.size(); i++){
49        cout << mas_2[i] << " ";
50    }
51    return 0;
52 }
```

Copyright © 2013-2024 - [algotester.com](#). All rights reserved. Algotester



About

Statistics

Help

Analysis

Events

BOHDANSAUKH

My Solutions

My teams

ARCHIVE

Problems

Rank

Solution Queue

Contact Us

College

Sponsors

Donate

english

Богдан Саук

```
9 cin >> n;
10 int mas[n];
11 vector<int> mas_0;
12 vector<int> mas_1;
13 vector<int> mas_2;
14
15 for (int i = 0; i < n; i++){
16     cin >> mas[i];
17     if (mas[i] % 3 == 0){
18         mas_0.push_back(mas[i]);
19     }
20     if (mas[i] % 3 == 1){
21         mas_1.push_back(mas[i]);
22     }
23     if (mas[i] % 3 == 2){
24         mas_2.push_back(mas[i]);
25     }
26 }
27
28 sort(mas_2.begin(), mas_2.end());
29 auto it = unique(mas_2.begin(), mas_2.end());
30 mas_2.erase(it, mas_2.end());
31
32 sort(mas_1.begin(), mas_1.end(), greater<int>());
33 auto it_1 = unique(mas_1.begin(), mas_1.end());
34 mas_1.erase(it_1, mas_1.end());
35
36 sort(mas_0.begin(), mas_0.end());
37 auto it_0 = unique(mas_0.begin(), mas_0.end());
38 mas_0.erase(it_0, mas_0.end());
39
40 cout << mas_0.size() + mas_1.size() + mas_2.size() << endl;
41
42 for (int i = 0; i < mas_0.size(); i++){
43     cout << mas_0[i] << " ";
44 }
45 for (int i = 0; i < mas_1.size(); i++){
46     cout << mas_1[i] << " ";
47 }
48 for (int i = 0; i < mas_2.size(); i++){
49     cout << mas_2[i] << " ";
50 }
51 return 0;
52 }
```

Copyright © 2013-2024 - [algotester.com](#). All rights reserved. Algotester

Час виконання ~ 30хв

Код без засобів STL:

About

Statistics

Help

Analysis

Events

BOHDANSIAKH

My Solutions

My teams

ARCHIVE

Problems

Rank

Solution Queue

Contact Us

College

Sponsors

Donate

english

Богдан Саук

Щоб вийти з повноекранного режиму, утримуйте клавішу Esc

```
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 void print_reverse_order(vector<int>& arr){
7     for (int i = arr.size() - 1; i >= 0; i--)
8         cout << arr[i] << " ";
9 }
10
11 void merge(vector<int>& arr, int left, int mid, int right){
12     int n1 = mid - left + 1;
13     int n2 = right - mid;
14     vector<int> L(n1), R(n2);
15
16     for (int i = 0; i < n1; i++)
17         L[i] = arr[left + i];
18     for (int j = 0; j < n2; j++)
19         R[j] = arr[mid + 1 + j];
20
21     int i = 0, j = 0;
22     int k = left;
23
24     while (i < n1 && j < n2) {
25         if (L[i] <= R[j]) {
26             arr[k] = L[i];
27             i++;
28         }
29         else {
30             arr[k] = R[j];
31             j++;
32         }
33         k++;
34     }
35
36     while (i < n1) {
37         arr[k] = L[i];
38         i++;
39         k++;
40     }
41
42     while (j < n2) {
43         arr[k] = R[j];
44         j++;
45         k++;
46     }
47 }
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Copyright © 2013-2024 - [algotester.com](#). All rights reserved.

Algotester

About

Statistics

Help

Analysis

Events

BOHDANSIAKH

My Solutions

My teams

ARCHIVE

Problems

Rank

Solution Queue

Contact Us

College

Sponsors

Donate

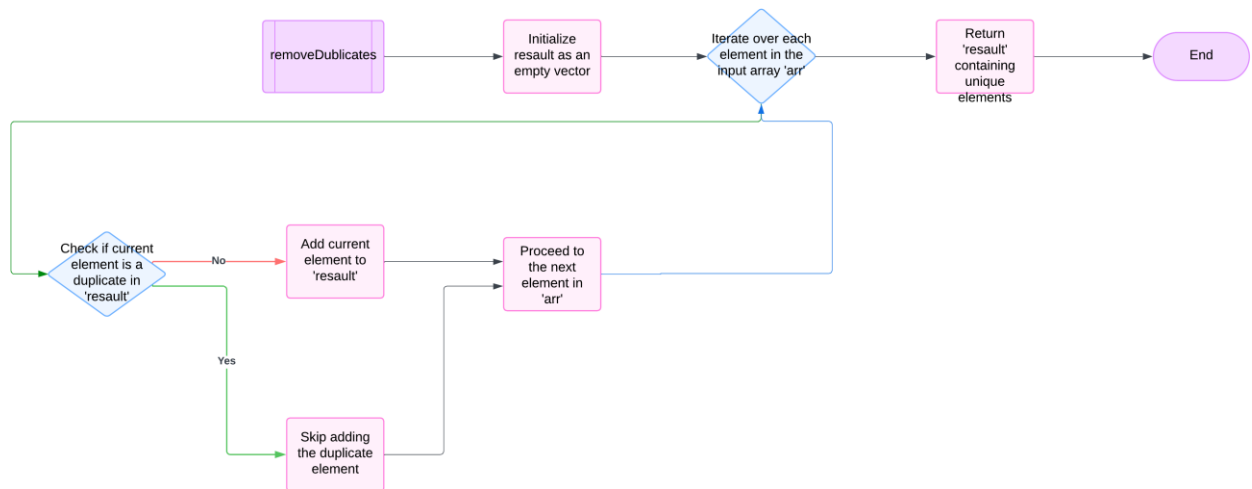
english

Богдан Саук

```
50
51 void mergeSort(vector<int>& arr, int left, int right){
52     if (left >= right)
53         return;
54
55     int mid = left + (right - left) / 2;
56     mergeSort(arr, left, mid);
57     mergeSort(arr, mid + 1, right);
58     merge(arr, left, mid, right);
59 }
60
61 void printVector(vector<int>& arr){
62     for (int i = 0; i < arr.size(); i++)
63         cout << arr[i] << " ";
64 }
65
66 vector<int> removeDuplicates(const vector<int>& arr){
67     vector<int> result;
68
69     for (int i = 0; i < arr.size(); i++){
70         bool is_duplicate = false;
71
72         for (int j = 0; j < result.size(); j++){
73             if (arr[i] == result[j]){
74                 is_duplicate = true;
75                 break;
76             }
77         }
78
79         if (!is_duplicate){
80             result.push_back(arr[i]);
81         }
82     }
83     return result;
84 }
85
86
87 int main(){
88     int n;
89     cin >> n;
90
91     int mas[n];
92     vector<int> mas_0;
93     vector<int> mas_1;
94     vector<int> mas_2;
95
96     for (int i = 0; i < n; i++){
97         int x;
98         cin >> x;
99         mas[i] = x;
100         if (i % 3 == 0) mas_0.push_back(x);
101         if (i % 3 == 1) mas_1.push_back(x);
102         if (i % 3 == 2) mas_2.push_back(x);
103     }
104
105     vector<int> res0 = removeDuplicates(mas_0);
106     vector<int> res1 = removeDuplicates(mas_1);
107     vector<int> res2 = removeDuplicates(mas_2);
108
109     printVector(res0);
110     printVector(res1);
111     printVector(res2);
112 }
```

Copyright © 2013-2024 - [algotester.com](#). All rights reserved.

Algotester



Завдання 5: Algotester Lab 6

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 sudoku.

Гра виглядає так:

Є поле розміром $N \times N$, в якому частина клітинок заповнена цифрами, а частина клітинок пусті (позначаються нулем). Також у нього є Q пар координат X та Y .

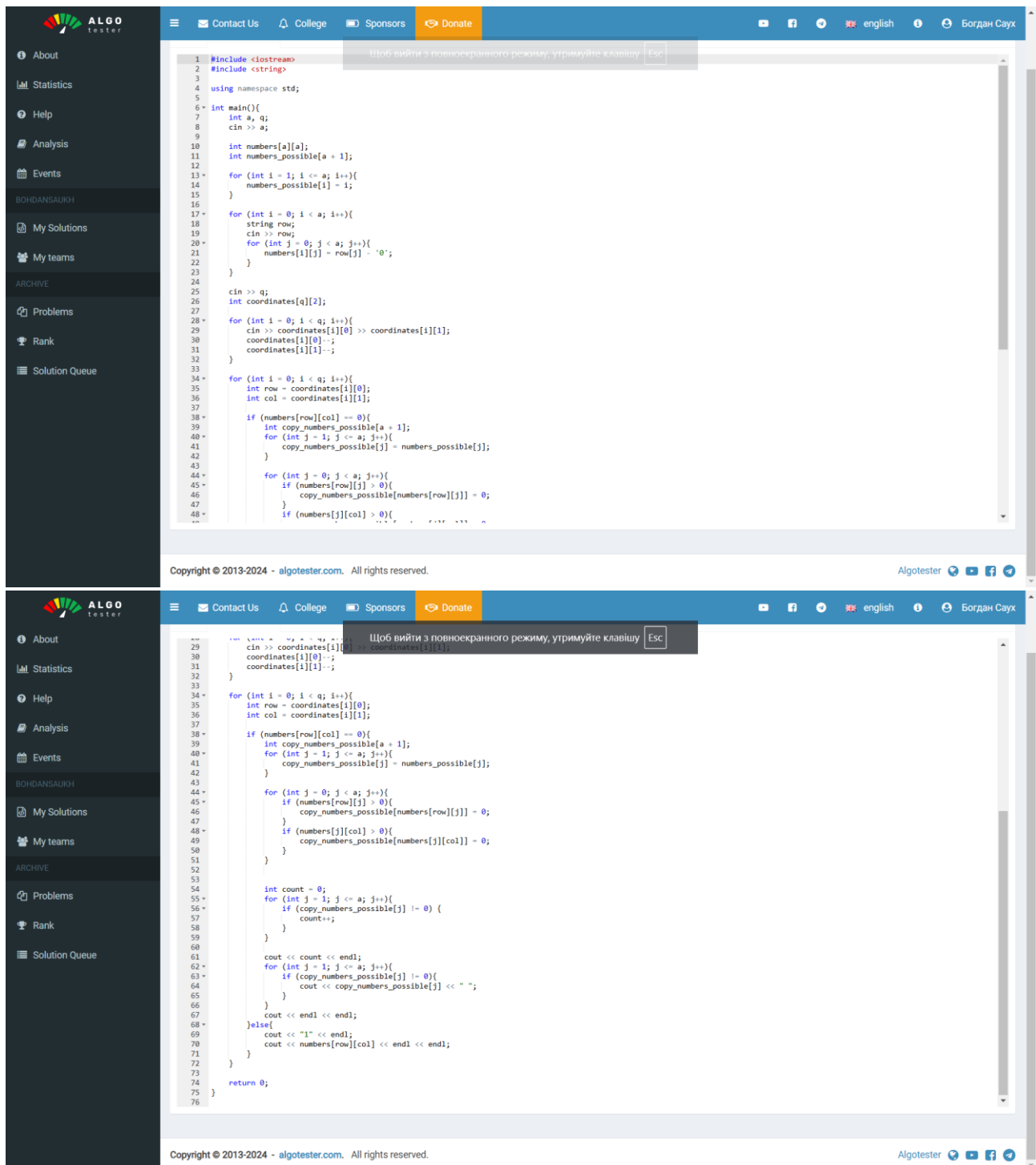
Завданням гри є написати до кожної координати скільки чисел туди можна вписати (якщо вона пуста) і які це числа (обов'язково в посортовані по зростанню!). В клітинку можна вписати лише ті числа, які не зустрічаються в рядку та стовбці, які перетинаються у цій клітинці.

Під час гри поле не міняється!

Також необов'язково, щоб це було валідне sudoku! Якщо є клітинка, в яку не можна вписати ніяку цифру - виведіть 0.

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

Код:



Час виконання ~ 50хв

Висновок:

У цьому епіку я навчився працювати з файлами, розібрався як працюють вектори і списки а також ознайомився із бібліотеками.

Pull request: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/596