

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



Звіт

про виконання лабораторних та практичних робіт блоку № 4

На тему: «Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.»

з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи №4

ВНС Лабораторної Роботи №5

Алготестер Лабораторної Роботи №2

Алготестер Лабораторної Роботи №3

Практичних Робіт до блоку №4

Виконав:

Студент групи ШІ-11
Саух Богдан Володимирович

Львів 2024

Тема роботи: Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами

Мета роботи: Навчитися ефективно працювати з одновимірними та двовимірними масивами, використовувати вказівники та посилання для оптимізації роботи з динамічними масивами, розібратися зі структурами даних і вкладеними структурами для зберігання складних даних, а також освоїти алгоритми обробки та маніпуляції масивами та структурами.

Теоретичні відомості:

1)Перелік тем:

1. Класи пам'яті у C++
2. Вступ до Масивів і Вказівників
3. Одновимірні Масиви
4. Вказівники та Посилання
5. Двовимірні Масиви
6. Динамічні Масиви
7. Структури Даних
8. Вкладені Структури
9. Використання структур
- 10.Алгоритми обробки та робота з масивами та структурами

2)Індивідуальний план опрацювання теорії:

1. **Класи пам'яті у C++**
 - [C++ Storage Classes](#)
 - [Memory Classes in C++](#)
2. **Вступ до Масивів і Вказівників**
 - [Introduction to Arrays in C++](#)
 - [Pointers in C++](#)
3. **Одновимірні Масиви**
 - [C++ One-Dimensional Arrays](#)
 - [Array Basics in C++](#)
4. **Вказівники та Посилання**
 - [Pointers and References in C++](#)
 - [References in C++](#)

5. Двовимірні Масиви

- [Two-Dimensional Arrays in C++](#)
- [2D Arrays in C++](#)

6. Динамічні Масиви

- [Dynamic Arrays in C++](#)
- [C++ Dynamic Memory](#)

7. Структури Даних

- [Structures in C++](#)
- [Introduction to Data Structures](#)

8. Вкладені Структури

- [Nested Structures in C++](#)
- [C++ Nested Structures](#)

9. Використання структур

- [C++ Structs and their Use](#)
- [Uses of Structures in C++](#)

10. Алгоритми обробки та робота з масивами та структурами

- [Algorithms for Arrays in C++](#)
- [Working with Arrays and Structures in C++](#)

Виконання роботи:

1) Перелік завдань:

- John Black - Epic 4 Task 1 - Theory Education Activities
- John Black - Epic 4 Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-8)
- John Black - Epic 4 Task 3 - Lab# programming: VNS Lab 4(варіант 22)
- John Black - Epic 4 Task 4 - Lab# programming: VNS Lab 5(варіант 22)

- John Black - Epic 4 Task 5 - Lab# programming: Algotester Lab 2(варіант 3)
- John Black - Epic 4 Task 6 - Lab# programming: Algotester Lab 3(варіант 3)
- John Black - Epic 4 Task 7 - Practice# programming: Class Practice Task
- John Black - Epic 4 Task 8 - Practice# programming: Self Practice Task
- John Black - Epic 4 Task 9 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)
- John Black - Epic 4 Task 10 - Results Evaluation and Release

2)Умови завдань:

Task 3:

Реалізувати з використанням масиву стек (перший прийшов, останній пішов), для чого організувати додавання, знищення елементів з масиву і друк масиву після кожної операції.

Task4: Використовуючи функції, розв'язати зазначене у варіанті завдання. Масив повинен передаватися у функцію як параметр.

У двовимірному масиві знайти середнє арифметичне першого стовпця й кількість елементів у кожному з наступних стовпців, що перевищують середнє арифметичне попереднього стовпця.

Task5:

Вам дано масив цілих чисел розміром N , на першій та останній клітинці розміщено по дрону.

Вони одночасно взлітають.

На початку кожного ходу швидкість дрону стає рівною значенню клітинки, у якій він знаходиться.

Тобто лівий дрон у першу секунду з клітинки з індексом 1 перелетить у клітинку з індексом a_1 , тобто його наступна позиція рахується як поточна позиція + число у поточній позиції (перегляньте пояснення для візуалізації) Правий робить аналогічно в протилежну сторону.

Вони це роблять до моменту, коли трапиться одна з зазначених подій:

Якщо 2 дрони опиняться в одній клітинці - ви виводите **Collision**.

Якщо лівий дрон опиниться справа від правого - це **Miss**

У випадку якщо вони зупиняться один навпроти одного, тобто у клітинках a_i та a_{i+1} - виведіть **Stopped**

Врахуйте, що перевіряти треба також до взльоту.

Task6:

Вам дана стрічка *s*.

Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

Task7: Перевірка чи слово або число є паліндромом

Задача

Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

Паліндром — це послідовність символів, яка однаково читається вперед і назад (наприклад, «radar», «level», «12321»).

Мета Задачі

Навчитися користуватися механізмами перевантаження функції та використовувати рекурсію для вирішення задач обчислення.

Вимоги:

1. Визначення функції:
 - a. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
 - a. `bool isPalindrome(const string& str, int start, int end);`
3. Перевантаження функцій:
 - a. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
 - b. `bool isPalindrome(ціле число);`
4. Рекурсія:
 - a. Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях.

Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок не перевищить кінець, після чого рядок буде визначено як паліндром.

Кроки реалізації

- Визначте та реалізуйте рекурсивну функцію *isPalindrome* для рядків.
- Визначте та реалізуйте перевантажену функцію *isPalindrome* для цілих чисел. Використати математичний підхід щоб перевірити чи число є паліндромом.

Task 8:

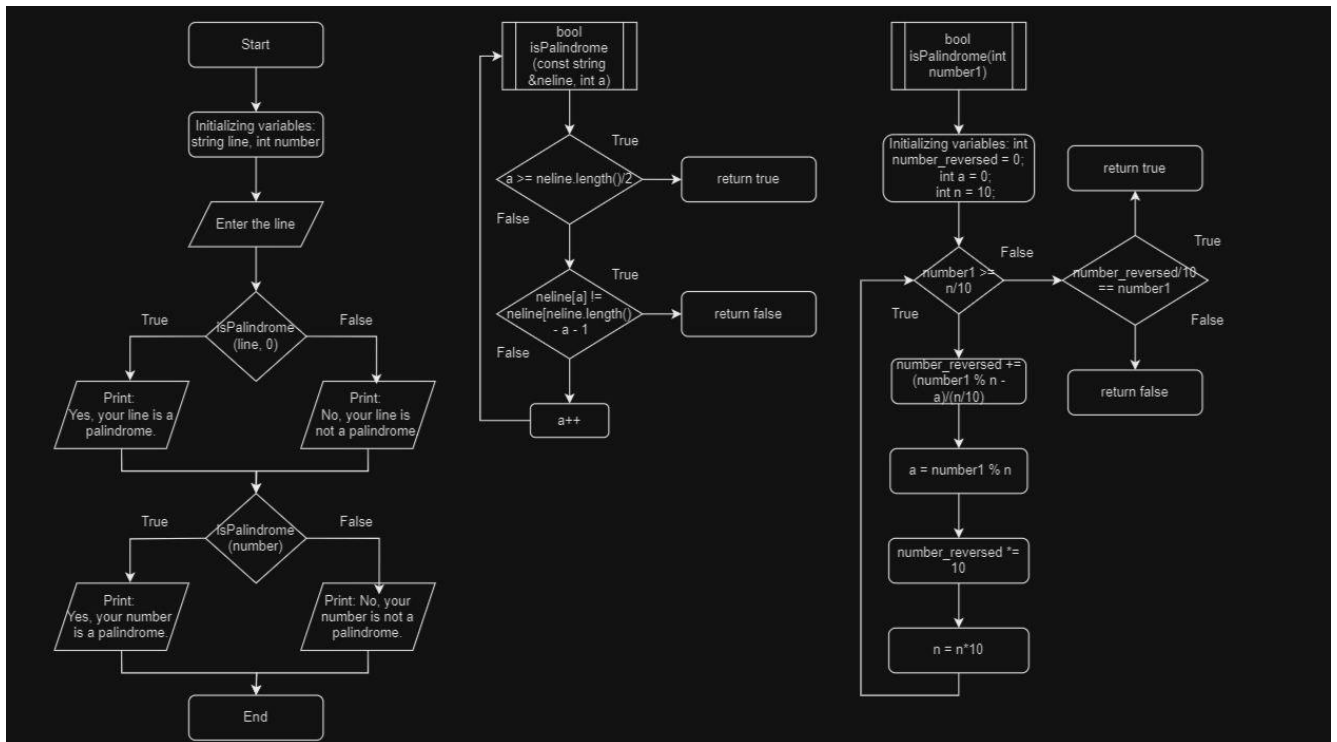
Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елемента.

Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

3)Дизайн та планова оцінка часу виконання завдань:

Task 7 - Practice# programming: Class Practice Task



Орієнтовний час виконання: 45хв

4)Код програм з посиланням на зовнішні ресурси:

Task 3 - Lab# programming: VNS Lab 4(варіант 22)

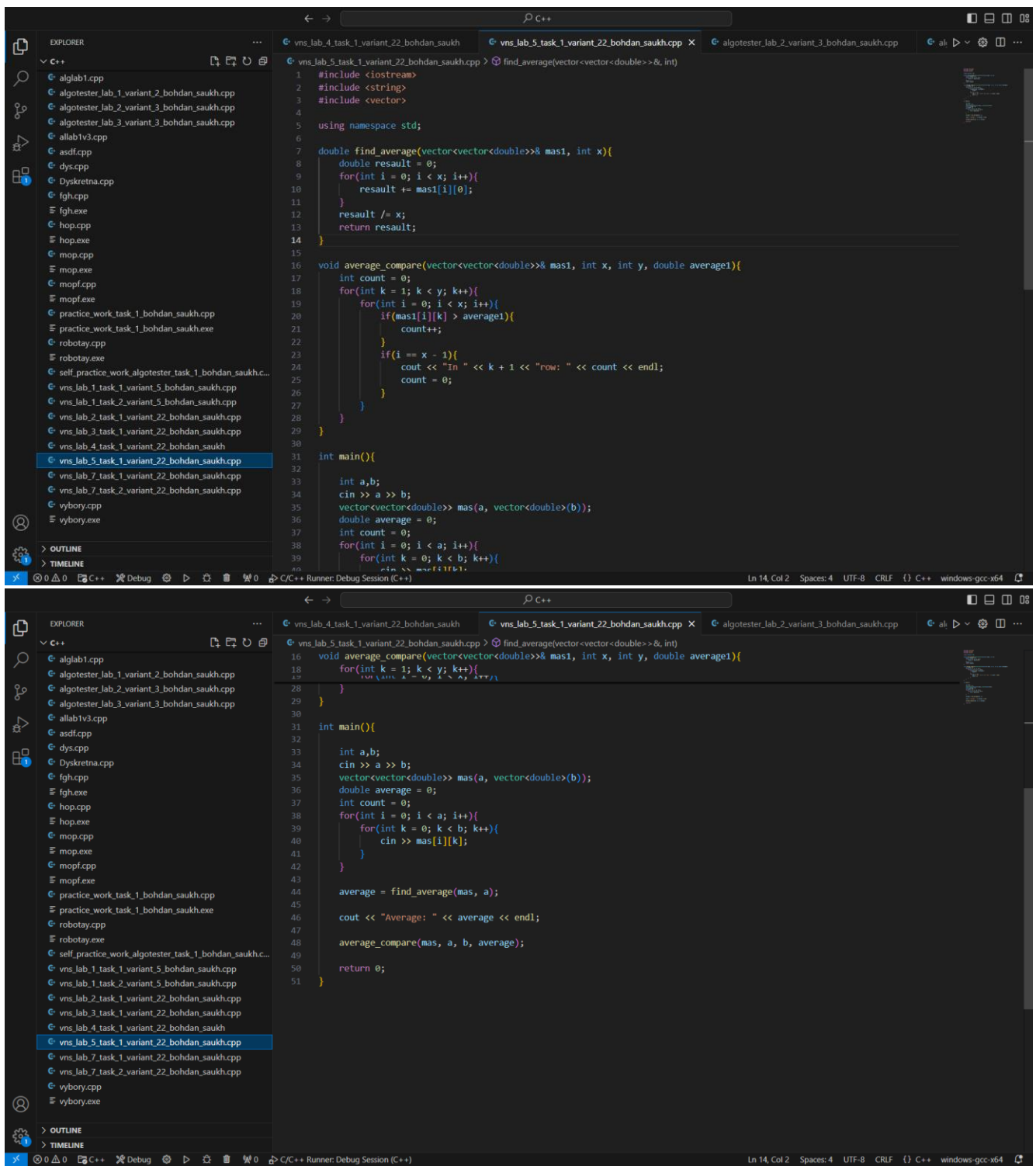
Посилання на файл програми:

```
1 #include <iostream>
2
3 using namespace std;
4
5 void push(int* mas1, int& top1, int element1, int n1){
6     if(top1 < n1 - 1){
7         top1++;
8         mas1[top1] = element1;
9         for(int i = 0; i <= top1; i++){
10             cout << mas1[i] << " ";
11         }
12     }else{
13         cout << "Not enough space!";
14     }
15 }
16
17 void destroy(int* mas1, int& top1, int n1){
18     if(top1 > -1){
19         mas1[top1] = 0;
20         top1--;
21         for(int i = 0; i <= top1; i++){
22             cout << mas1[i] << " ";
23         }
24     }else{
25         cout << "Stack is already empty!";
26     }
27 }
28
29 void print(int mas1[], int top1){
30     for(int i = 0; i <= top1; i++){
31         cout << mas1[i] << " ";
32     }
33 }
34
35 int main(){
36     int n;
37     int choose = -2;
38     cout << "Enter stack size: ";
39     cin >> n;
40     int* mas1 = new int[n];
41     int top = -1;
42     int element;
43     do{
44         cout << endl << "Choose an option: add(1), destroy(0), print(2) or exit(-1): ";
45         cin >> choose;
46         if(choose == 1){
47             cout << "Enter an element: ";
48             cin >> element;
49             push(mas1, top, element, n);
50         }
51         if(choose == 0){
52             destroy(mas1, top, n);
53         }
54         if(choose == 2){
55             print(mas1, top);
56         }
57     } while (choose != -1);
58     return 0;
59 }
```

```
29 void print(int mas1[], int top1){
30     for(int i = 0; i <= top1; i++){
31         cout << mas1[i] << " ";
32     }
33 }
34
35 int main(){
36     int n;
37     int choose = -2;
38     cout << "Enter stack size: ";
39     cin >> n;
40     int mas[n];
41     int top = -1;
42     int element;
43     do{
44         cout << endl << "Choose an option: add(1), destroy(0), print(2) or exit(-1): ";
45         cin >> choose;
46         if(choose == 1){
47             cout << "Enter an element: ";
48             cin >> element;
49             push(mas, top, element, n);
50         }
51         if(choose == 0){
52             destroy(mas, top, n);
53         }
54         if(choose == 2){
55             print(mas, top);
56         }
57     } while (choose != -1);
58     return 0;
59 }
```

Task 4 - Lab# programming: VNS Lab 5(варіант 22)

Посилання на файл програми:



Task 5 - Lab# programming: Algotester Lab 2(варіант 3)

Посилання на файл програми:


```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6     int n;
7     cin >> n;
8     int res1 = 0;
9     int res2 = n - 1;
10    int road[n];
11
12    for (int i = 0; i < n; i++){
13        cin >> road[i];
14    }
15
16    while (! (res1 >= (res2 - 1))){
17        res1 = res1 + road[res1];
18        res2 = res2 - road[res2];
19    }
20
21    cout << res1 + 1 << " " << res2 + 1 << endl;
22
23    if (res1 == res2){
24        cout << "Collision";
25    }
26
27    if (res1 == (res2 - 1)){
28        cout << "Stopped";
29    }
30
31    if (res1 > res2){
32        cout << "Miss";
33    }
34
35    return 0;
36 }
```

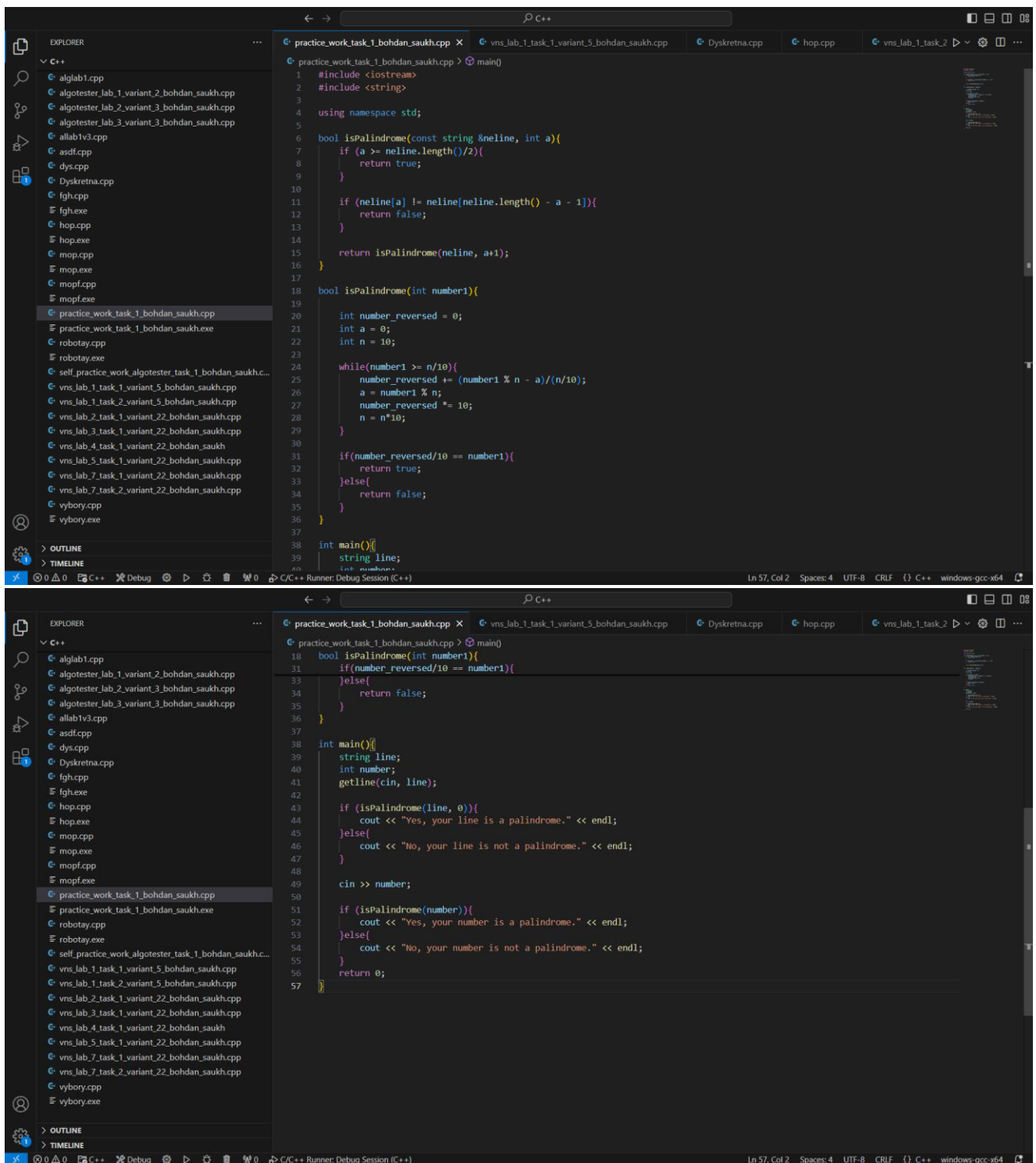
Task 6 - Lab# programming: Algotester Lab 3(варіант 3)

Посилання на файл програми:

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main(){
7     string line;
8     string resault;
9     getline(cin, line);
10    int count = 1;
11    resault.push_back(line[0]);
12    for(int i = 1; i < line.length(); i++){
13        if(line[i-1] == line[i]){
14            count++;
15        }
16        if(line[i-1] != line[i]){
17            if(count > 1){
18                resault += to_string(count);
19            }
20            resault.push_back(line[i]);
21            count = 1;
22        }
23        if(i == line.length() - 1){
24            if(count > 1){
25                resault += to_string(count);
26            }
27        }
28    }
29    cout << resault;
30    return 0;
31 }
```

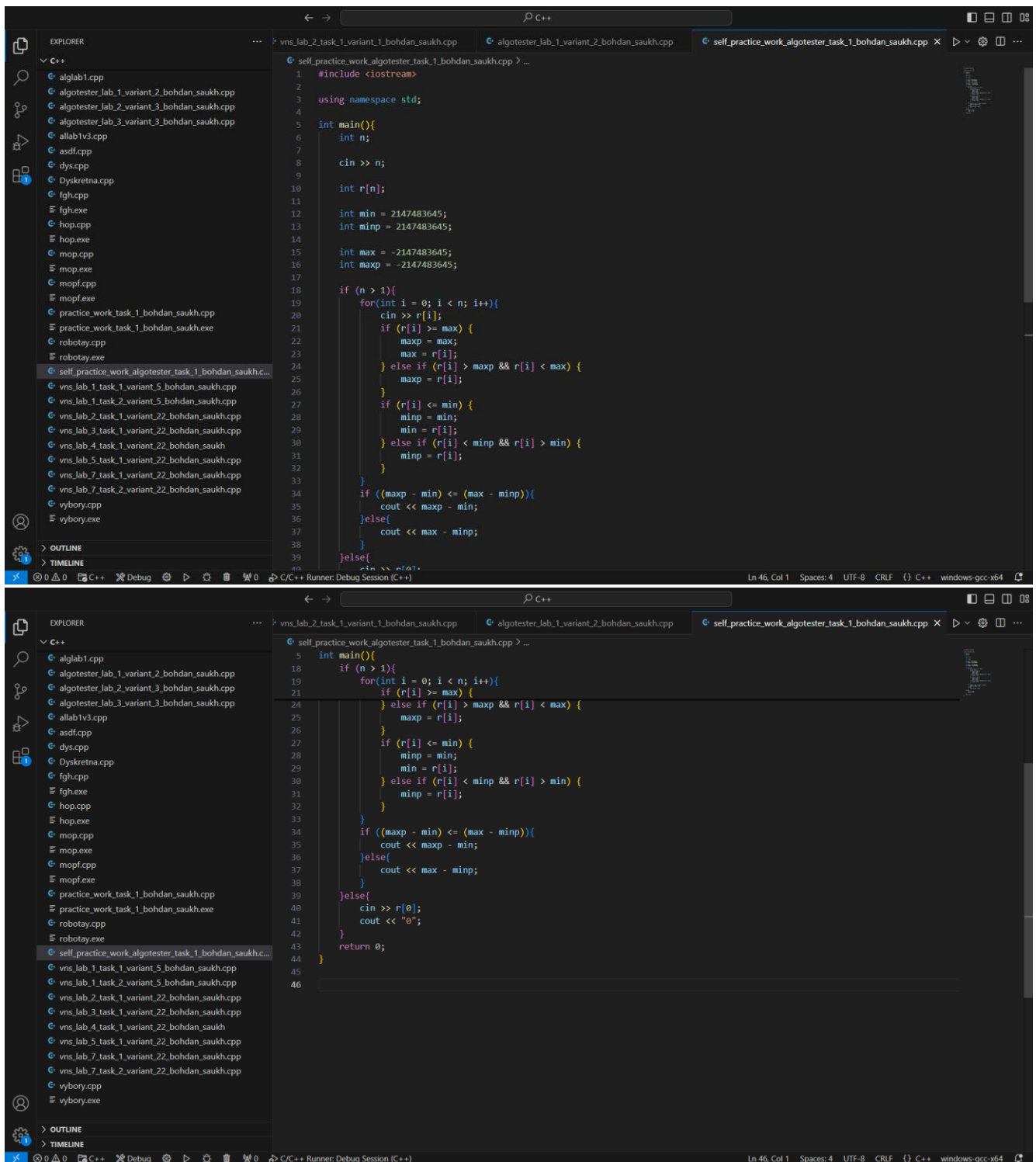
Task 7 - Practice# programming: Class Practice Task

Посилання на файл програми:



Task 8 - Practice# programming: Self Practice Task

Посилання на файл програми:



5)Результати виконання завдань та фактично затрачений час

Task 3 - Lab# programming: VNS Lab 4(варіант 22)

```

Enter stack size: 2

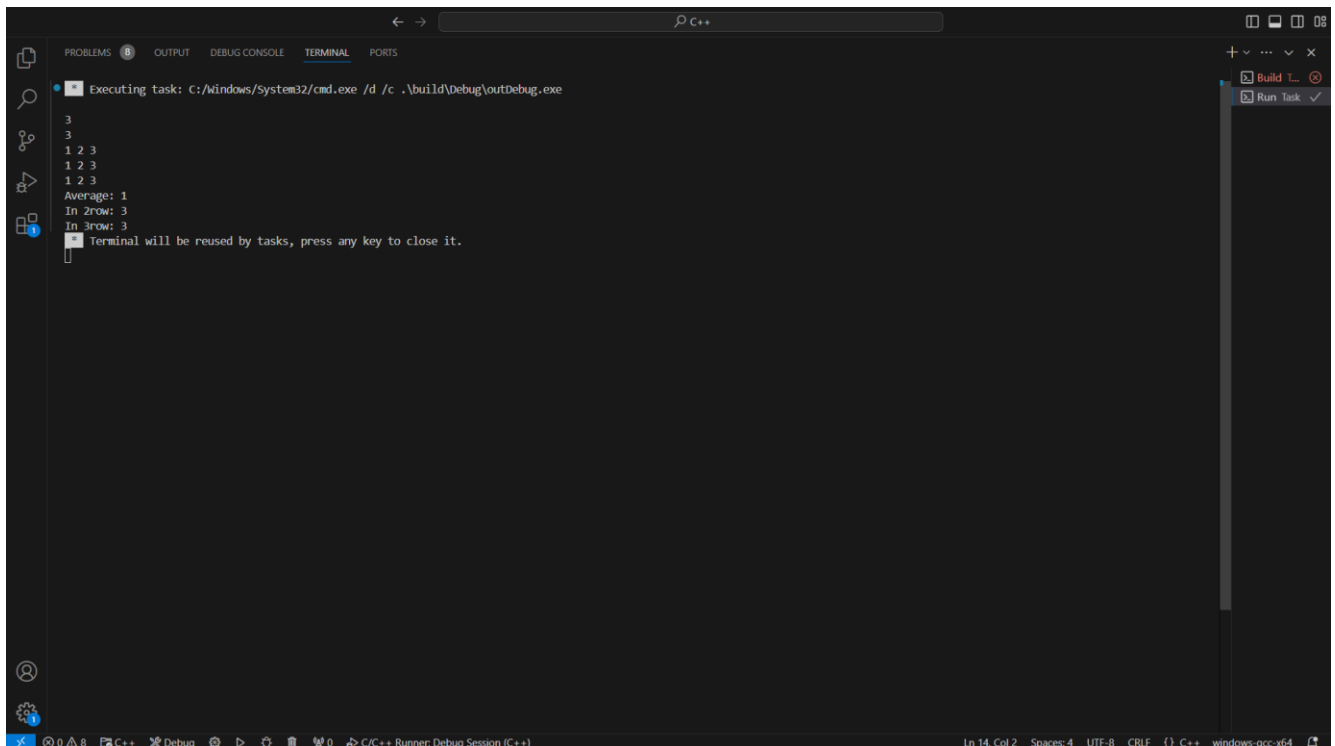
Choose an option: add(1), destroy(0), print(2) or exit(-1): 1
Enter an element: 2
2
Choose an option: add(1), destroy(0), print(2) or exit(-1): 1
Enter an element: 4
2 4
Choose an option: add(1), destroy(0), print(2) or exit(-1): 1
Enter an element: 4
Not enough space!
Choose an option: add(1), destroy(0), print(2) or exit(-1): 0
2
Choose an option: add(1), destroy(0), print(2) or exit(-1): 0

Choose an option: add(1), destroy(0), print(2) or exit(-1): 0
Stack is already empty!
Choose an option: add(1), destroy(0), print(2) or exit(-1): -1

```

Фактично затрачений час: 1 год

Task 4 - Lab# programming: VNS Lab 5(варіант 22)



The screenshot shows a C++ IDE with a terminal window. The terminal output is as follows:

```

Executing task: C:/windows/system32/cmd.exe /d /c .\build\Debug\outDebug.exe
3
3
1 2 3
1 2 3
1 2 3
Average: 1
In 2row: 3
In 3row: 3
Terminal will be reused by tasks, press any key to close it.

```

The IDE interface includes a sidebar with icons for Explorer, Search, Run and Debug, and Source Control. The top bar shows the file path and the C++ language. The bottom status bar indicates the current line and column (Ln 14, Col 2) and the encoding (UTF-8).

Фактичний час затрачений на виконання: 50хв

Task 5 - Lab# programming: Algotester Lab 2(варіант 3)

The screenshot shows the Algotester Lab 2 interface. On the left is a dark sidebar with navigation links: About, Statistics, Help, Analysis, Events, My Solutions, My teams, Problems, Rank, and Solution Queue. The main area has a top navigation bar with links to Contact Us, College, Sponsors, and Donate. Below this is a compiler selection dropdown set to 'C++ 23' and a file input field containing 'main.cpp'. A code editor area is mostly empty, with a 'Submit' button at the bottom. A table below the editor shows the submission results:

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
4 days ago	C++ 23	Accepted	0.003	1.066	View

Showing 1 to 1 of 1 rows

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

Фактичний час затрачений на виконання: 35хв

Task 6 - Lab# programming: Algotester Lab 3(варіант 3)

The screenshot shows the Algotester Lab 3 interface, similar to Lab 2 but with a different set of submissions. The sidebar and top navigation are the same. The submission table is as follows:

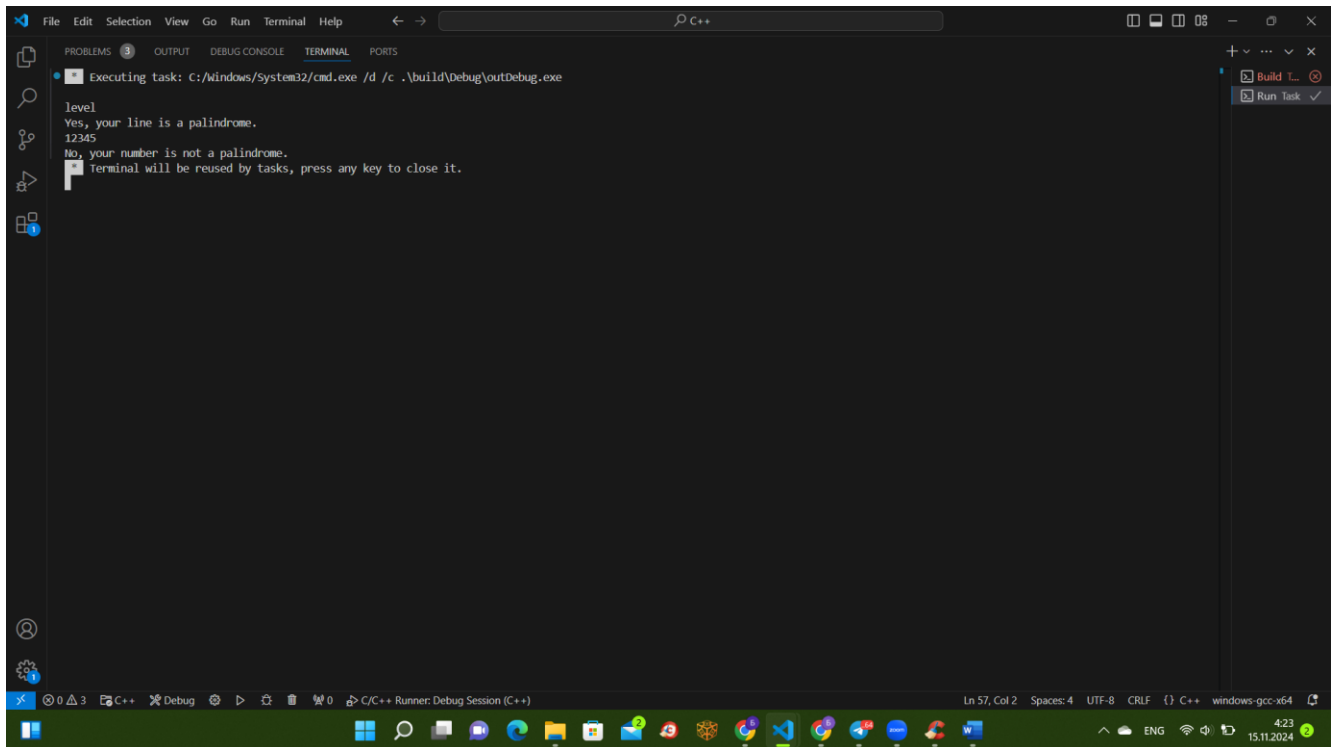
Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
3 days ago	C++ 23	Accepted	0.003	1.113	View
3 days ago	C++ 23	Wrong Answer 4	0.003	0.957	View
4 days ago	C++ 23	Wrong Answer 1	0.003	0.949	View

Showing 1 to 3 of 3 rows

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

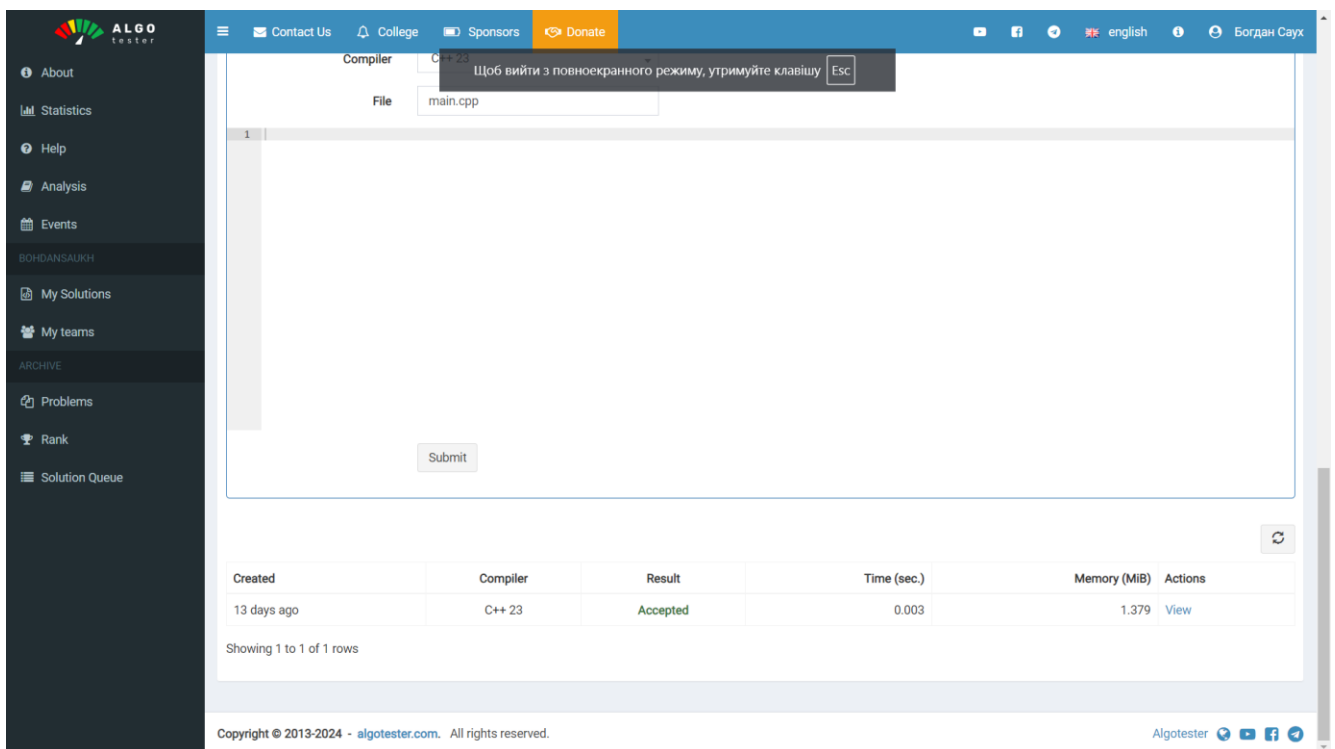
Фактичний час затрачений на виконання: 2 год

Task 7 - Practice# programming: Class Practice Task



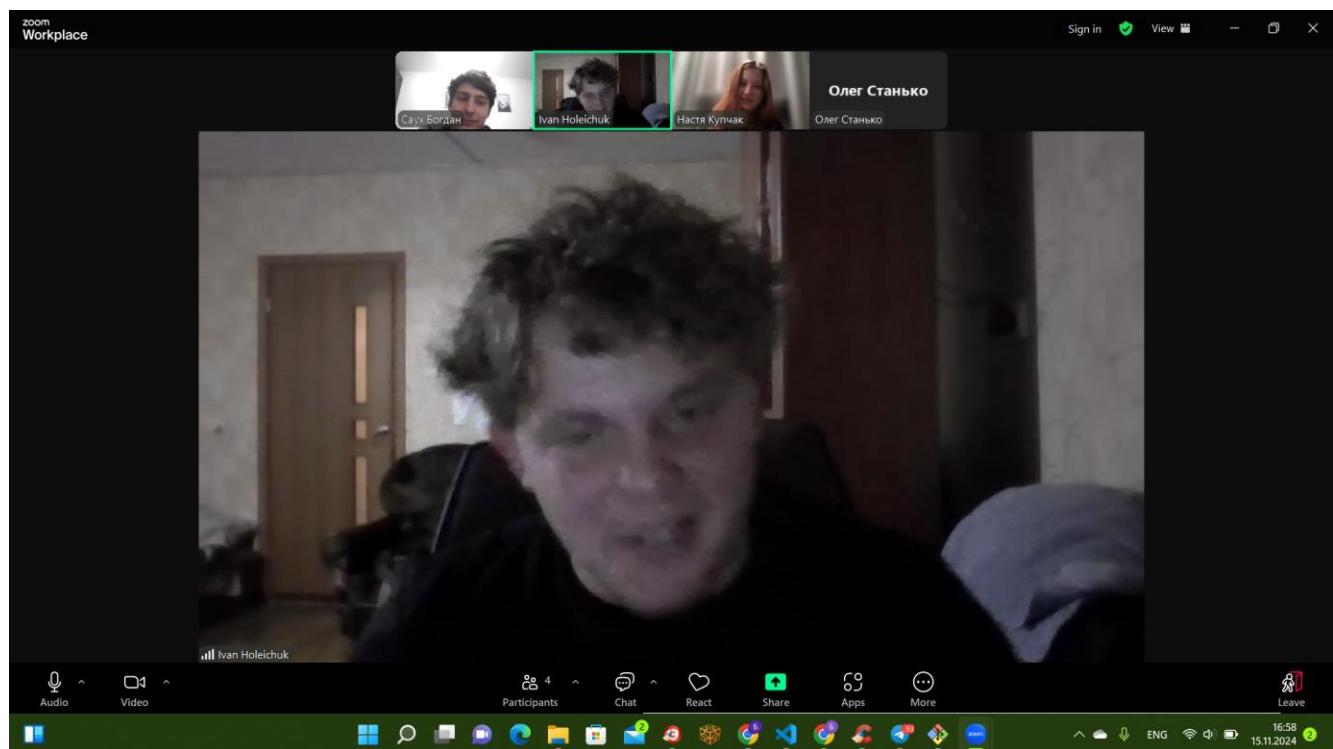
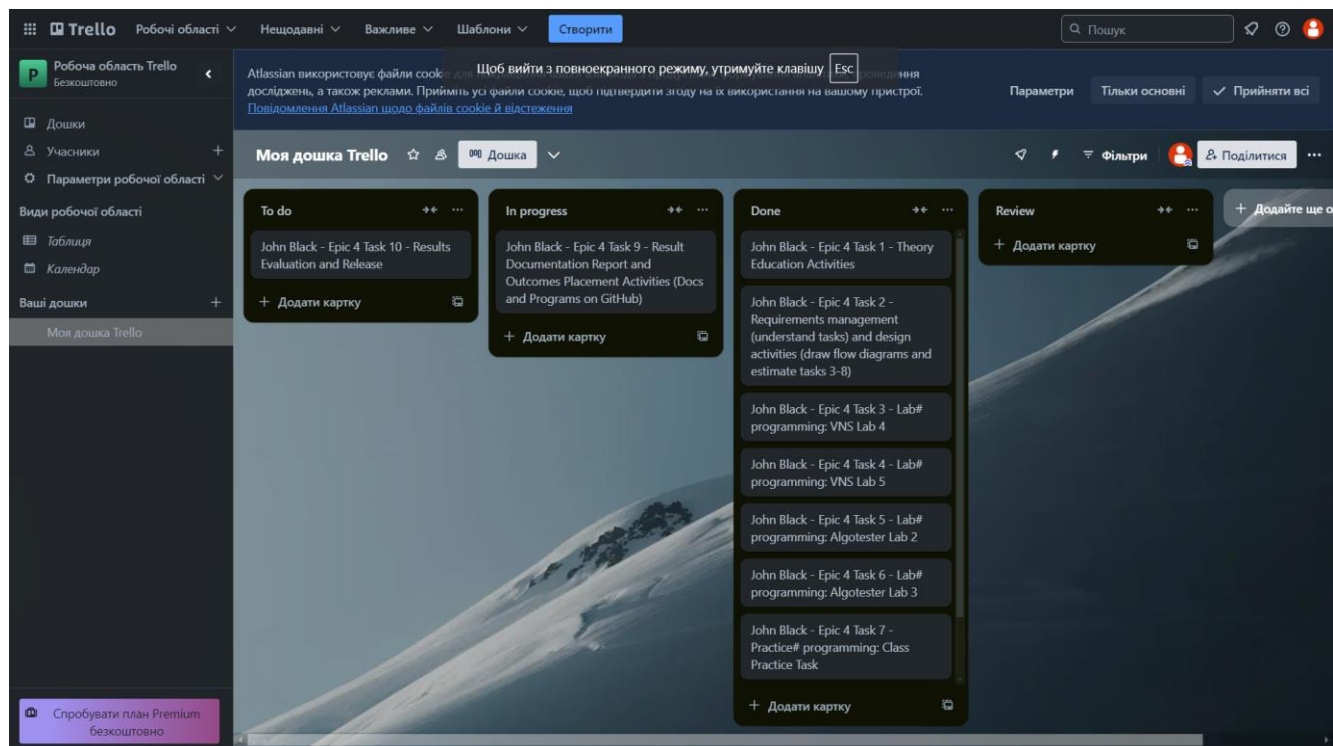
Фактичний час виконання: 1 год

Task 8 - Practice# programming: Self Practice Task



Фактичний час виконання: 40хв

6) Робота з комадою



Висновок: Опановуючи роботу з масивами різних типів, вказівниками, посиланнями, динамічними структурами даних та алгоритмами їх обробки, ми набуваємо необхідних знань і навичок для ефективного управління даними та

їхньої оптимізації. Це дозволяє будувати більш гнучкі та продуктивні програми, здатні працювати з великими обсягами даних, зберігаючи при цьому структурованість і логічну цілісність інформації.

Посилання на пул реквест: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/315