

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:

Практичних Робіт до блоку № 4

Виконав:

Студент групи ІІІ-13
Мурашко Владислав Сергійович

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 працювати з одновимірними, двовимірними та динамічними масивами, вказівниками та посиланнями, структурами та вкладеними структурами.

Теоретичні відомості:

1. Теми, необхідні для виконання роботи:

- Одновимірні та двовимірні масиви
- Динамічні масиви
- Структури та вкладені структури
- Вказівники та посилання

2. Джерела використані для ознайомлення з вищезазначеними темами:

- З усією необхідною інформацією я був знайомий з власного досвіду.

Виконання роботи:

1) Опрацювання завдання та вимог до програми та середовища

Завдання №1 - Class Practice Work – Перевірка чи слово або число є паліндромом

Задача

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

Мета Задачі

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

Вимоги:

1. Визначення функції:

Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.

2. Приклад визначення функції:

bool isPalindrome(const string& str, int start, int end);

3. Перевантаження функцій:

Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.

bool isPalindrome(ціле число);

4. Рекурсія:

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.

Завдання №2 - Self Practice Work – Algotester lab 2 variant 1

Задача:

У вас є дорога, яка виглядає як N чисел.

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?

Input

У першому рядку ціле число N - кількість чисел

У другому рядку масив r , який складається з N цілих чисел

Output

Єдине ціле число m - мінімальна втома, яку можна отримати

Завдання №3 - Self Practice Work – Algotester lab 3 variant 1

Задача:

Ви з'явилися у світі під назвою Атод посеред Пустелі Безправ'я. Так сталося, що Ви попали саме в той час і місце, де ведеться битва між чаклункою Ліною і темними силами, які хочуть знищити цей світ. На жаль, трапилась халепа, бо деякі слова із книги чар були пошкоджені під час битви. Одне таке слово можна відновити виконавши ритуал зцілення над пошкодженими буквами. Ритуал зцілення можна виконати на всіх **підряд розташованих пошкоджених** буквах. Вам не залишається нічого іншого як допомогти Ліні відновити ці слова і сказати скільки мінімально треба провести таких ритуалів, щоб прочитати одне з наймогутніших у цьому світі заклять - Поневолення Дракона!

Input

У першому рядку N - кількість рядків у заклятті.

В наступних N рядках - набір слів w_1, \dots, w_m , розділених пробілами, де кожне слово може містити малі латинські літери та символ $\#$, який позначає пошкоджену букву.

Output

Єдине ціле число - мінімальна кількість ритуалів, які потрібно провести, щоб відновити заляття.

Завдання №4 - Self Practice Work – Algotester lab 3 variant 2

Задача:

Вам дано 2 масиви розміром N та M . Значення у цих масивах унікальні.

Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

Input

У першому рядку ціле число N

у другому рядку N цілих чисел $a_1..a_n$

У третьому рядку ціле число M

у четвертому рядку M цілих чисел $b_1..b_n$

Output

У першому рядку одне ціле число - кількість елементів, які наявні в обох масивах одночасно.

У другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об'єднанням двох даних).

Завдання №5 – VNS Lab 4 - Task 1-10

Задача:

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити 5 перших елементів масиву.
- 4) Додати в кінець масиву 3 нових елементи.
- 5) Роздрукувати отриманий масив.

Завдання №6 - VNS Lab 5 - Task 1-10

Задача:

Написати функцію, що перевіряє чи є від'ємні елементи в зазначеному рядку двовимірного масиву. Знищити з масиву всі рядки

з від'ємними елементами, знищений рядок заповнюється 0 і переноситься в кінець масиву.

Завдання №7 – Algotester lab 2 variant 2

Задача:

У вас є масив g розміром N . Також вам дано 3 цілих числа.

Спочатку ви маєте видалити з масиву ці 3 числа, які вам дані. Після цього перетворити цей масив у масив сум, розміром $N_{\text{new}}-1$ (розмір нового масиву після видалення елементів), який буде відображати суми сусідніх елементів нового масиву.

Далі необхідно вивести масив сум на екран.

Input

У першому рядку ціле число N - кількість чисел

У другому рядку масив g , який складається з N цілих чисел

У третьому рядку 3 цілих числа, a, b, c які треба видалити з масиву

Output

У першому рядку ціле число M - кількість чисел у масиві, який буде виведено

У наступному рядку M чисел - новий масив

Завдання №8 – Algotester lab 3 variant 3

Задача:

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.

Input

У першому рядку стрічка S

Output

Стрічка $S_{\text{compressed}}$

2) Дизайн та планована оцінка часу виконання завдань

Завдання №1 - Class Practice Work - Менеджмент бібліотеки

Дизайн алгоритму завдання знаходиться в папці “ Diagrams ” під назвою
practice_work_task_1_vladyslav_murashko.drawio

Попередньо розраховував, що це завдання займе 30 хвилин

Завдання №2 - Self Practice Work – Algotester lab 2 variant 1

Попередньо розраховував, що це завдання займе 1 годину

Завдання №3 - Self Practice Work – Algotester lab 3 variant 1

Попередньо розраховував, що це завдання займе 30 хв

Завдання №4 - Self Practice Work – Algotester lab 3 variant 1

Попередньо розраховував, що це завдання займе 30 хв

Завдання №5 – VNS Lab 4 - Task 1-10

Попередньо розраховував, що це завдання займе 30 хв

Завдання №6 – VNS Lab 5 - Task 1-10

Попередньо розраховував, що це завдання займе 30 хв

Завдання №7 – Algotester lab 2 variant 2

Попередньо розраховував, що це завдання займе 30 хв

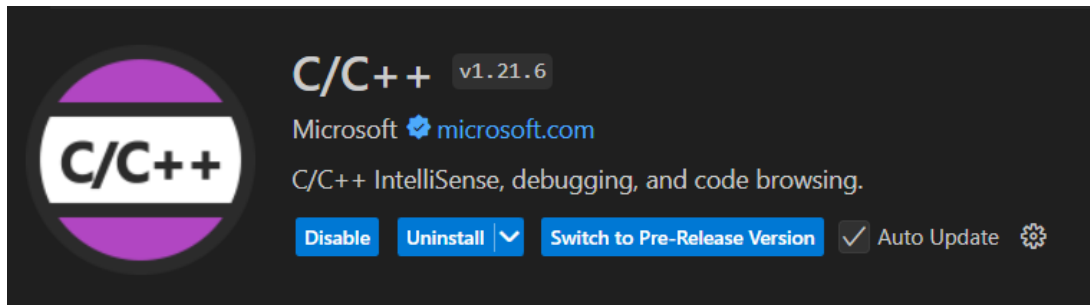
Завдання №8 – Algotester lab 3 variant 3

Попередньо розраховував, що це завдання займе 30 хв

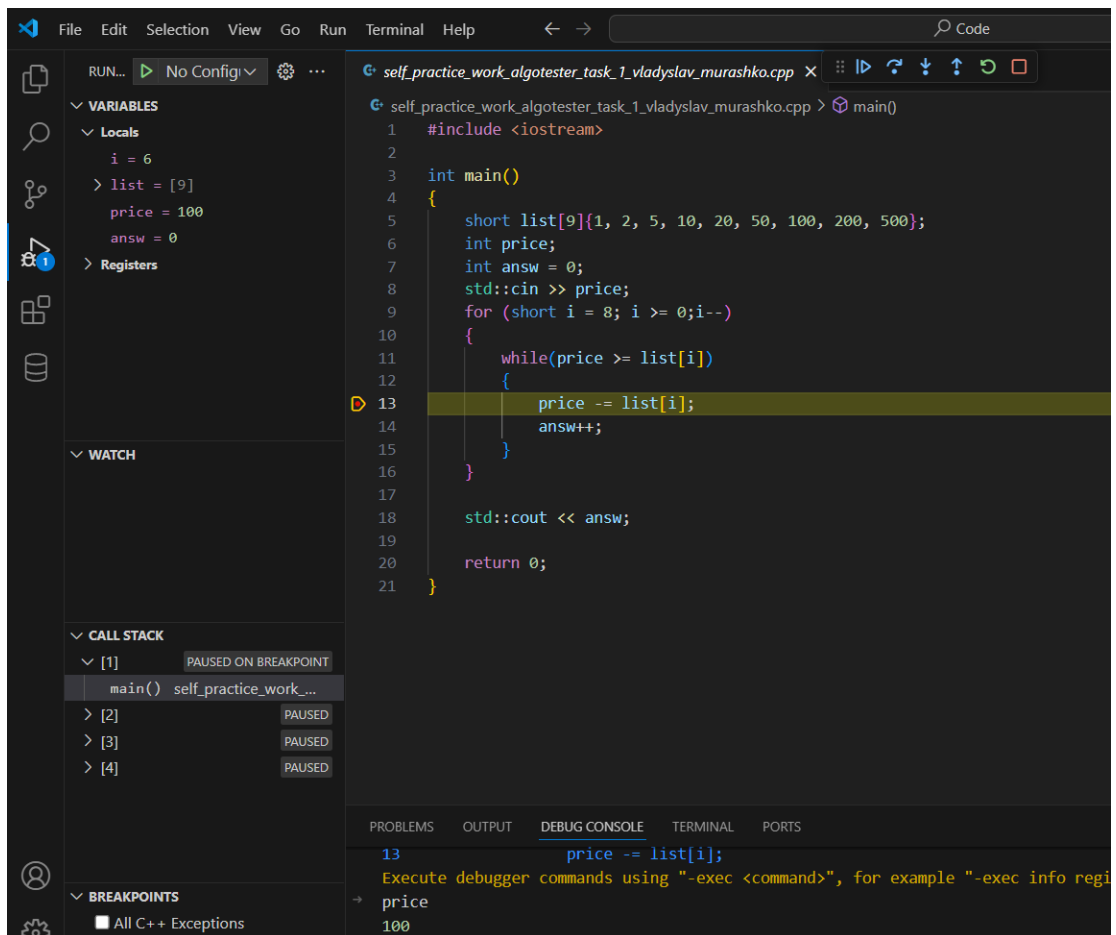
3) Конфігурація середовища для виконання завдань

```
* epic_4_practice_and_labs_vladyslav_murashko
main
```

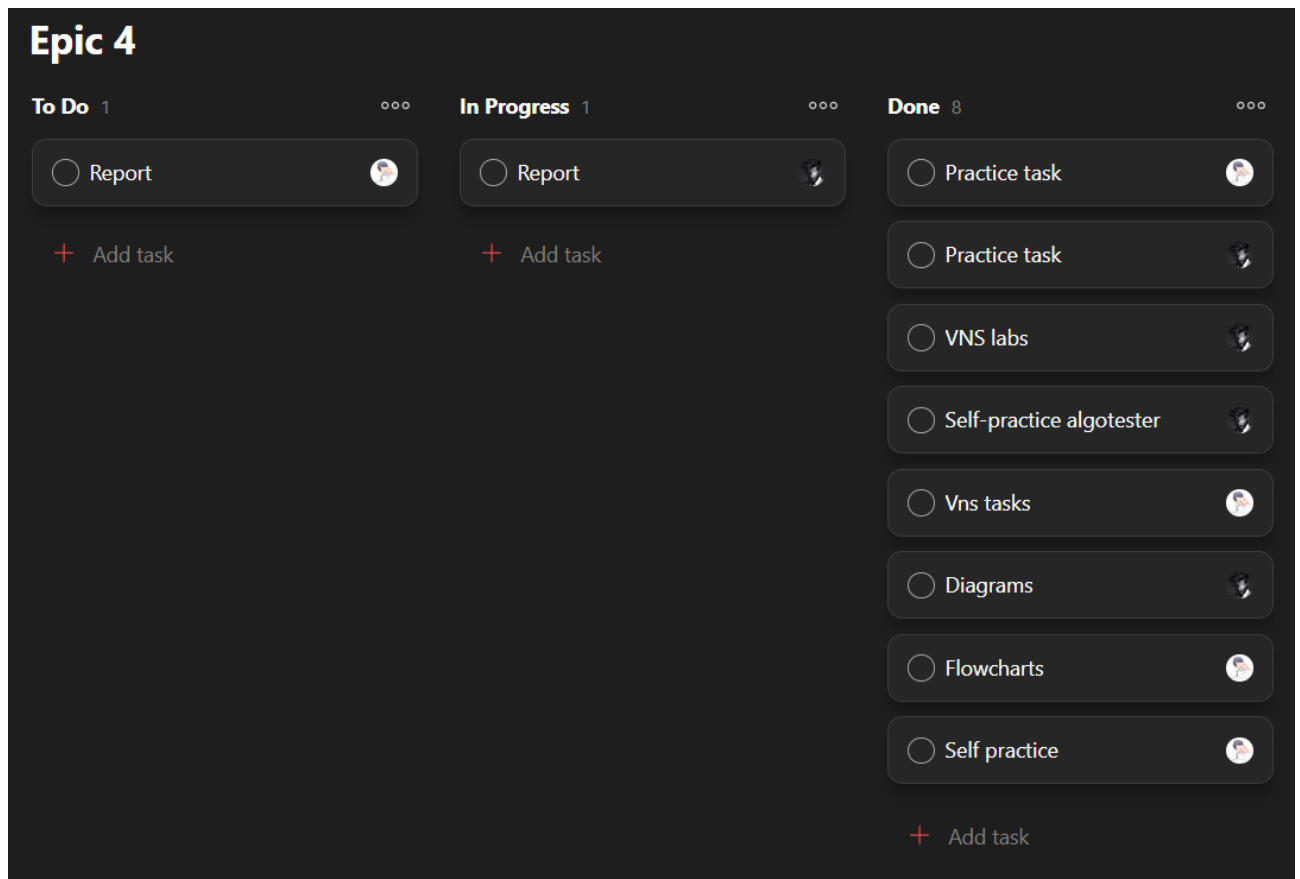
Git та моя гілка



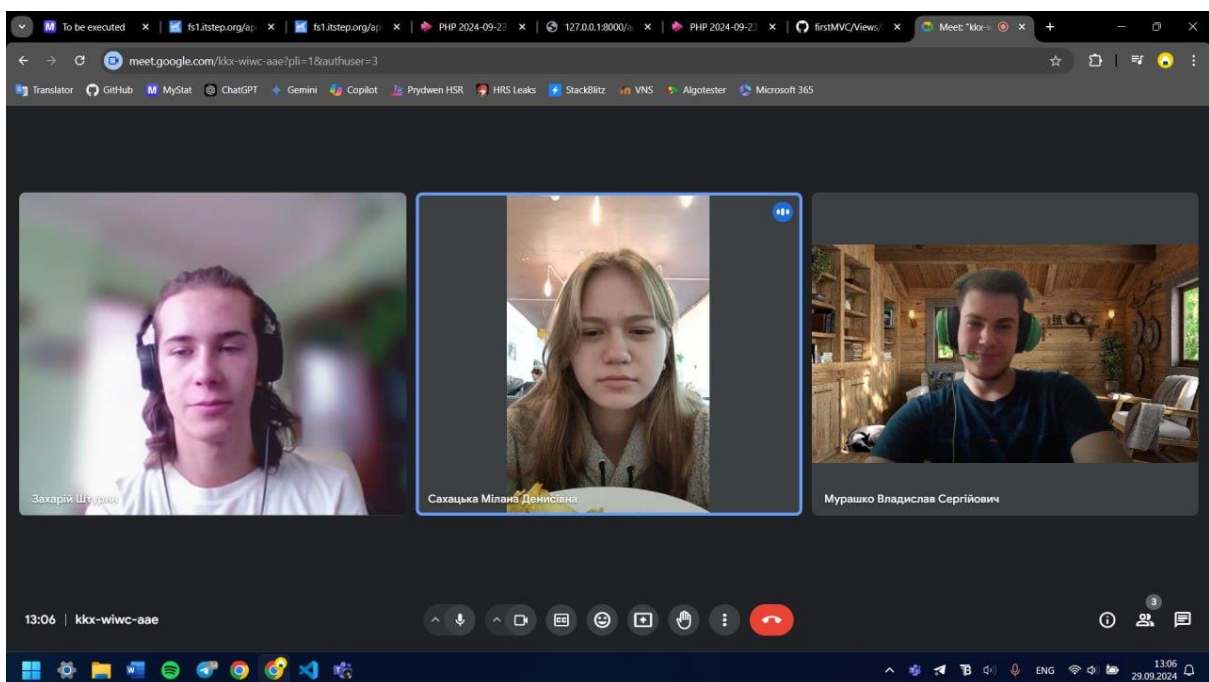
Встановлене розширення C/C++ для VS Code



Налаштований дебагер



Командна дошка



Скрін зустрічі нашої команди

4) Код програми з посиланням на зовнішні ресурси

- До завдання №1 код `practice_work_task_1_vladyslav_murashko.cpp`
- До завдання №2 код `self_practice_work_algotester_task_1_vladyslav_murashko.cpp`
- До завдання №3 код `self_practice_work_algotester_task_2_vladyslav_murashko.cpp`
- До завдання №4 код `self_practice_work_algotester_task_3_vladyslav_murashko.cpp`
- До завдання №5 код `vns_lab_4_task_1_variant_10_vladyslav_murashko.cpp`
- До завдання №6 код `vns_lab_5_task_1_variant_10_vladyslav_murashko.cpp`
- До завдання №7 код `algotester_lab_2_variant_2_vladyslav_murashko.cpp`
- До завдання №8 код `algotester_lab_3_variant_3_vladyslav_murashko.cpp`

5) Результати виконаних завдань, тестування та фактично затрачений час

Завдання №1 - Class Practice Work - Менеджмент бібліотеки

```
Cheking string "HiiH" on palindrome: 1
Cheking number "12333321" on palindrome: 1

Cheking string "HsiiH" on palindrome: 0
Cheking number "122333321" on palindrome: 0
```

Зайняло часу – 1 година

Завдання №2 - Self Practice Work – Algotester lab 2 variant 1

5	5
1 2 2 4 4	1 1 2 4 4
2	3

Зайняло часу – 30 хвилин

Завдання №3 - Self Practice Work – Algotester lab 3 variant 1

3	
Hel## worl#	
My name is ####	1
Bye wo##d!	Wery impor##nt mes#ag#e
4	3

Зайняло часу – 30 хвилин

Завдання №4 - Self Practice Work – Algotester lab 3 variant 2

5	3
1 2 3 4 5	5 4 3
5	5
4 5 6 7 8	1 2 3 4 8
2	2
8	6

Зайняло часу – 30 хвилин

Завдання №5 – VNS Lab 4 - Task 1-10

```
Initial array:

41 18467 6334 26500 19169 15724 11478 29358 26962 24464

Resulting array:

15724 11478 29358 26962 24464 5705 28145 23281
```

Зайняло часу – 30 хвилин

Завдання №6 – VNS Lab 5 - Task 1-10

```
Initial matrix:

-35 91 58 24 93
48 2 82 86 88
29 69 5 51 85
-85 -81 66 51 60
15 28 26 -23 -84

Matrix after deleting rows with negative numbers:

48 2 82 86 88
29 69 5 51 85
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

Зайняло часу – 30 хвилин

Завдання №7 – Algotester lab 2 variant 2

5	5
1 2 3 4 5	1 2 3 4 5
4 5 6	1 2 3
2	1
3 5	9

Зайняло часу – 30 хвилин

Завдання №8 – Algotester lab 3 variant 3

AAAABBBccccG	bbbNAN
A4B2c4G	b3NAN

Зайняло часу – 30 хвилин

Висновок: Я навчився працювати з одновимірними, двовимірними та динамічними масивами, вказівниками та посиланнями, структурами та вкладеними структурами.