Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**Звіт**

**про виконання розрахунково-графічних робіт блоку № 7**

***з дисципліни:*** «Основи програмування»

до:

ВНС Розрахунково-графічних робіт № 1-4

Практичних Робіт до блоку № 7

**Виконав:**

Студент групи ШІ-11

Вербицький Юрій Віталійович

Львів 2024

**Тема роботи:** Змінні, константи, умовні оператори та розгалуження, масиви, цикли, оператори, функції, робота з файлами, введення та виведення даних, структури, вказівники.

**Мета роботи:** Узагальнити знання та навички з мов C/C++ здобуті за час курсу «Основ програмування».

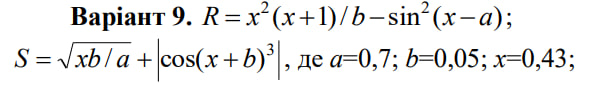
**Виконання роботи:**

**1)Перелік завдань:**

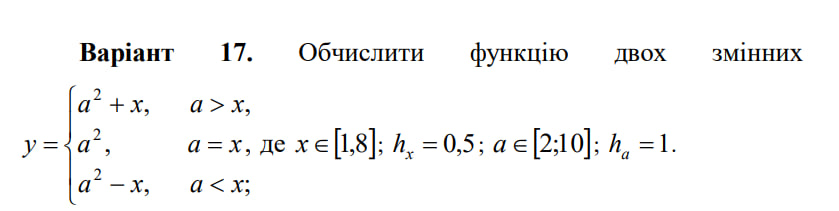
* Task 1: VNS practice work 1 task 9
* Task 2: VNS practice work 2 task 17
* Task 3: VNS practice work 3 task 14
* Task 4: VNS practice work 4 task 12
* Task 5, 6, 7, 8: Algotester practice work

**2)Умови завдань:**

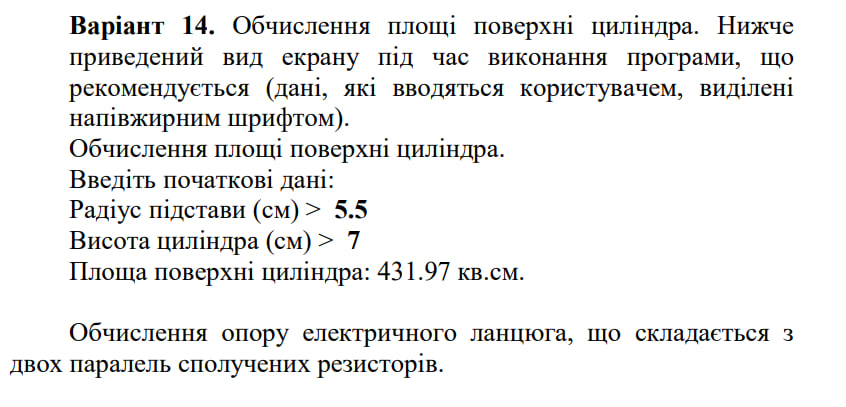
**Task 1:** Розробити лінійний алгоритм для розв’язання задачі.



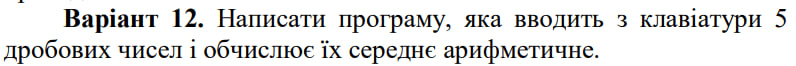
**Task 2:** Розробити алгоритм, що розгалужується для розв’язання задачі номер якої відповідає порядковому номеру студента в журналі викладача



**Task 3:** Написати програму згідно свого варіанту.



**Task 4:** Написати програму згідно свого варіанту



**Task 5: Algotester practice Lab1v2**У вас є стіл, у якого є 4 ніжки, довжини яких вам дано.

Ви хочете зробити ніжки рівної довжини, для цього ви відпиляєте dd від кожної ніжки (тобто вам буде дано 4 числа, кожне з яких буде означати відпилювання від відповіної ніжки стола).

Якщо під час відпилювання найдовша ніжка стола буде у 2 рази більша-рівна ніж найменша ніжка - стіл перевернеться, але відпилювати ніжки це вам не завадить.

Тобто якщо hmax>=2∗hminhmax>=2∗hmin то стіл перевертається. Увага, це може статися і між початком та кінцем відпилювання, наприклад коли відпиляють 2, але ще не встигнуть відпиляти 3тю ніжку. Також ми вважаємо що перед відпилюванням стіл не перевернеться.

Ваше завдання сказати чи після усіх маніпуляцій стіл буде цілий та паралельний підлозі.

Якщо довжина, яку відріжуть буде більша за довжину ножки - вам треба вивести ERROR.

Увага! Навіть якщо стіл перевернеться - ви все одно відпилюєте ніжки і можете отримати ERROR.

Input

4 цілих числа h1,2,3,4- довжини ніжок стола 4 цілих числа d1,2,3,4 - довжина, яку відпиляють від відповідної ножки

Output

YES - якщо стіл буде стояти паралельно площині підлоги та довжина найменшої ніжки не буде рівна нулю.

ERROR - у випадку якщо ви відпиляєте більшу довжину ніж має ножка

NO - у інших випадках

**Task 5: Self practice** Вам необхідно розробити програму, яка виконує наступні дії:  
Зчитує текстовий файл input.txt. Визначає кількість слів у файлі, кількість рядків та кількість символів (включаючи пробіли). Зберігає результати аналізу у файл output.txt у форматі:  
Копіювати код  
Кількість рядків: X  
Кількість слів: Y  
Кількість символів: Z

Знаходить усі слова, які починаються з великої літери, і записує їх у файл capitalized\_words.txt, кожне слово з нового рядка.

Вхідні дані:

Текстовий файл input.txt, який містить декілька рядків тексту.

Вихідні файли:

Файл output.txt із результатами аналізу.

Файл capitalized\_words.txt із усіма словами, що починаються з великої літери.

**Task 6: Algotester practice Lab2v1**

У вас є дорога, яка виглядає як NN чисел.

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

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 NN - кількість чисел

У другому рядку масив rr, який складається з NN цілих чисел

Output

Єдине ціле число mm - мінімальна втома, яку можна отримати

**Task 6: Self practice**

Завдання: Аналіз шахової дошки

Розробіть програму для роботи з шаховою дошкою розміром 8x8, яка заповнена значеннями 0 або 1. Програма повинна виконувати наступні дії:

Ввід даних про шахову дошку:

Користувач вводить значення для кожної клітини шахової дошки.

Кожне значення має бути або 0 (порожня клітина), або 1 (заповнена клітина).

Якщо користувач вводить неправильне значення, програма повинна вивести повідомлення про помилку і попросити повторити введення для тієї ж клітини.

Аналіз шахової дошки:

Підрахувати кількість заповнених клітин (1) у кожному рядку.

Визначити координати всіх заповнених клітин (у форматі "рядок, стовпець").

Обчислити загальну кількість заповнених клітин на всій дошці.

Вивід результатів:

Для кожного рядка вивести кількість заповнених клітин.

Вивести список координат усіх заповнених клітин.

Вивести загальну кількість заповнених клітин на дошці.

**Task 7: Algotester practice Halloween**

До Зеника і Марічки на Хелловін завітав їхній старий друг Андрій. Офіційна причина його візиту — дізнатися про справи друзів та просто побалакати з ними. Але Зенику і Марічці добре відомі справжні наміри Андрія — він просто хоче дістати від друзів цукерок.

У Зеника і Марічки є по одному мішку цукерок. Усього в мішку Зеника є n цукерок, причому ii-та із них коштує ai гривень. Аналогічно, у мішку Марічки є mm цукерок, і ціна ii-ї рівна bi гривень. Вони хочуть дати Андрію дві цукерки — одну з мішка Зеника, а іншу — з мішка Марічки.

Будучи доволі жадібними, воно готові віддати Андрію лише найдешевші цукерки. Тобто, як і з першого мішка, так і з другого, вони виберуть по найдешевшій цукерці. Якою буде вартість отриманого Андрієм подарунка? Іншими словами, знайдіть сумарну вартість двох цукерок, які отримає Андрій.

Вхідні дані

У першому рядку задано два цілих числа n та m — кількості цукерок в мішках Зеника та Марічки відповідно.

У другому рядку задано n цілих чисел, розділених пробілами. Вони описують вартості Зеникових цукерок.

У третьому рядку аналогічно описані mm цукерок Марічки.

Вихідні дані

У єдиному рядку виведіть одне ціле число — вартість подарунку, який отримає Андрій.

**Task 7: Self practice**

**Реалізувати програму для обчислення добутку двох матриць**

**Task 8: Algotester practice Lab3v3**

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 SS

Output

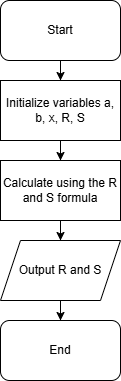
Стрічка Scompressed

**Task 8: Self practice**

Розробити програму, яка дозволяє ввести оцінки та обчислити середній бал. Програма також повинна вивести максимальну і мінімальну оцінку

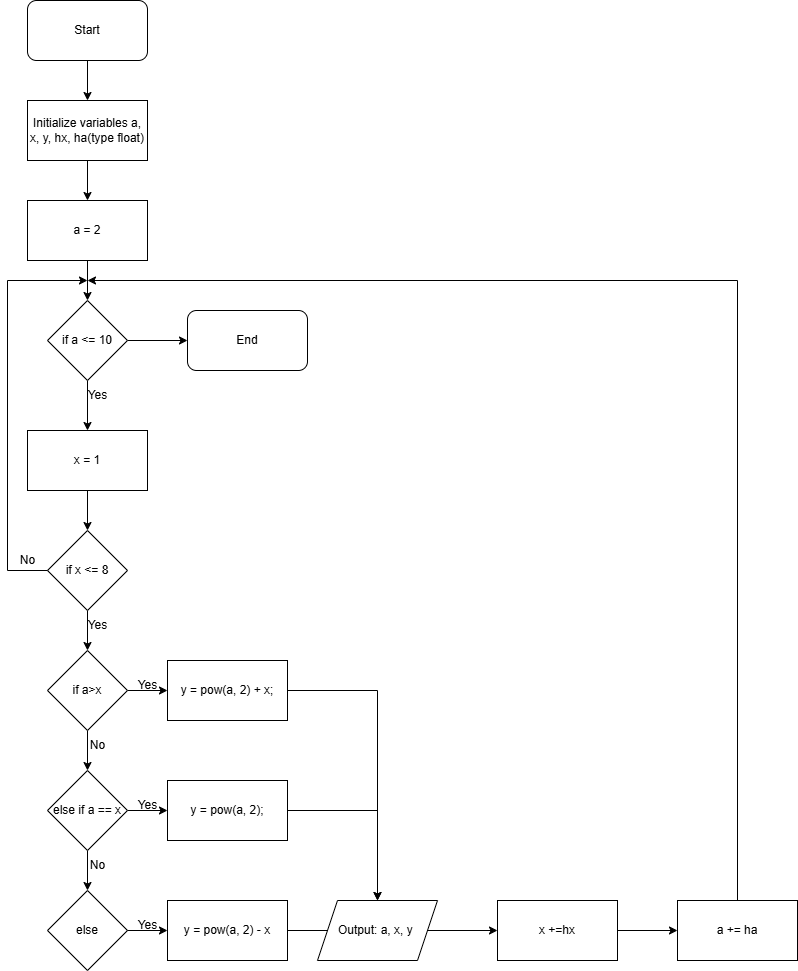
**3)Дизайн та планова оцінка часу виконання завдань:**

Task 1: VNS practice work 1 task 9



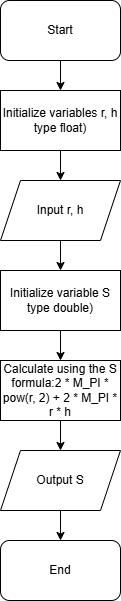
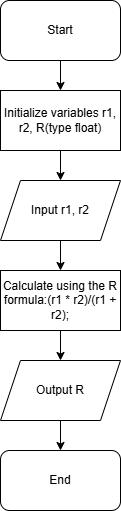
Орієнтовний час виконання: 5хв

Task 2: VNS practice work 2 task 17



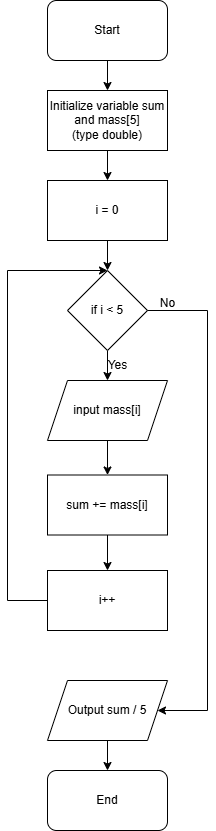
Орієнтовний час виконання: 25хв

Task 3: VNS practice work 3 task 14

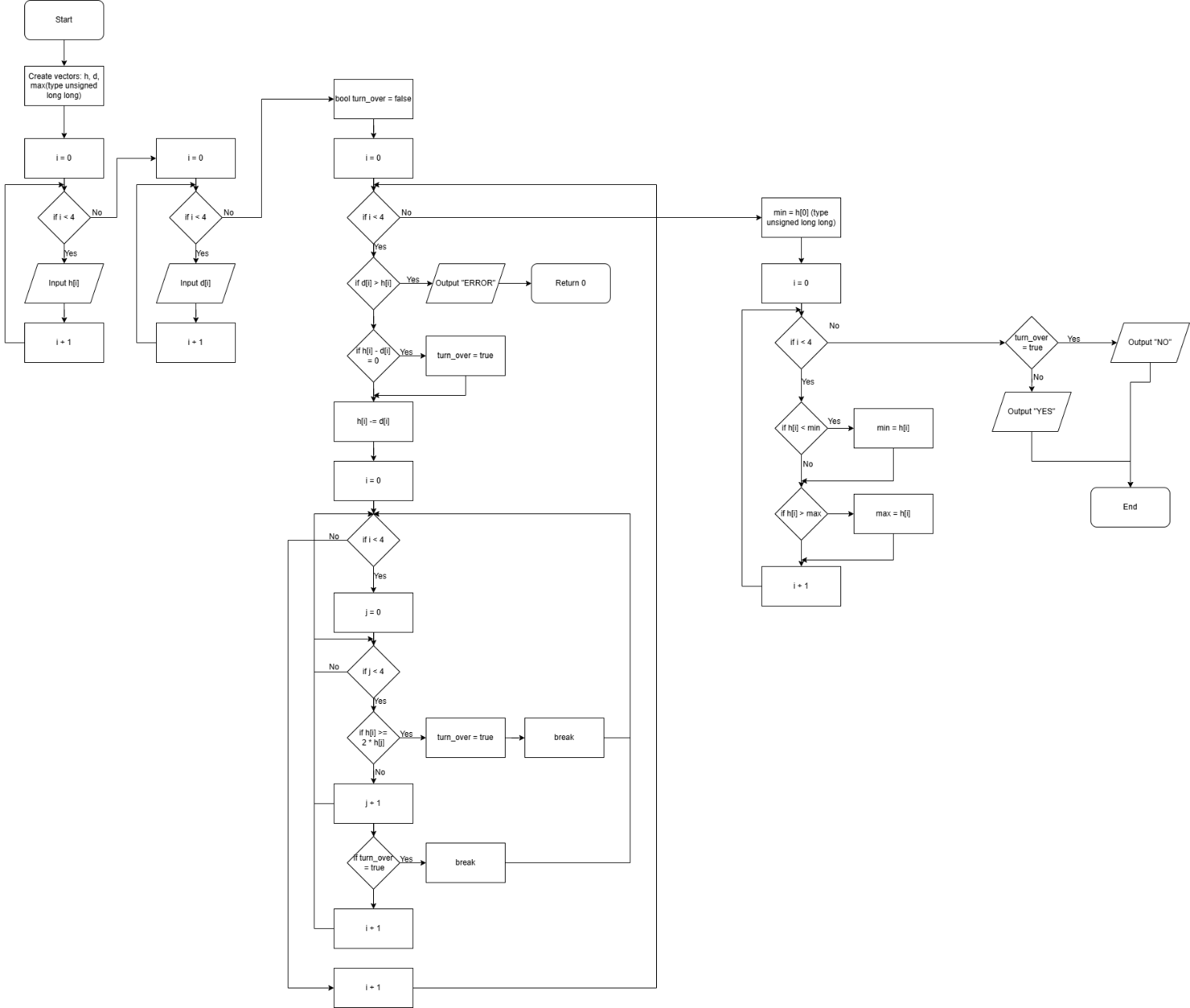
Орієнтовний час виконання: 15хв

Task 4: VNS practice work 4 task 12

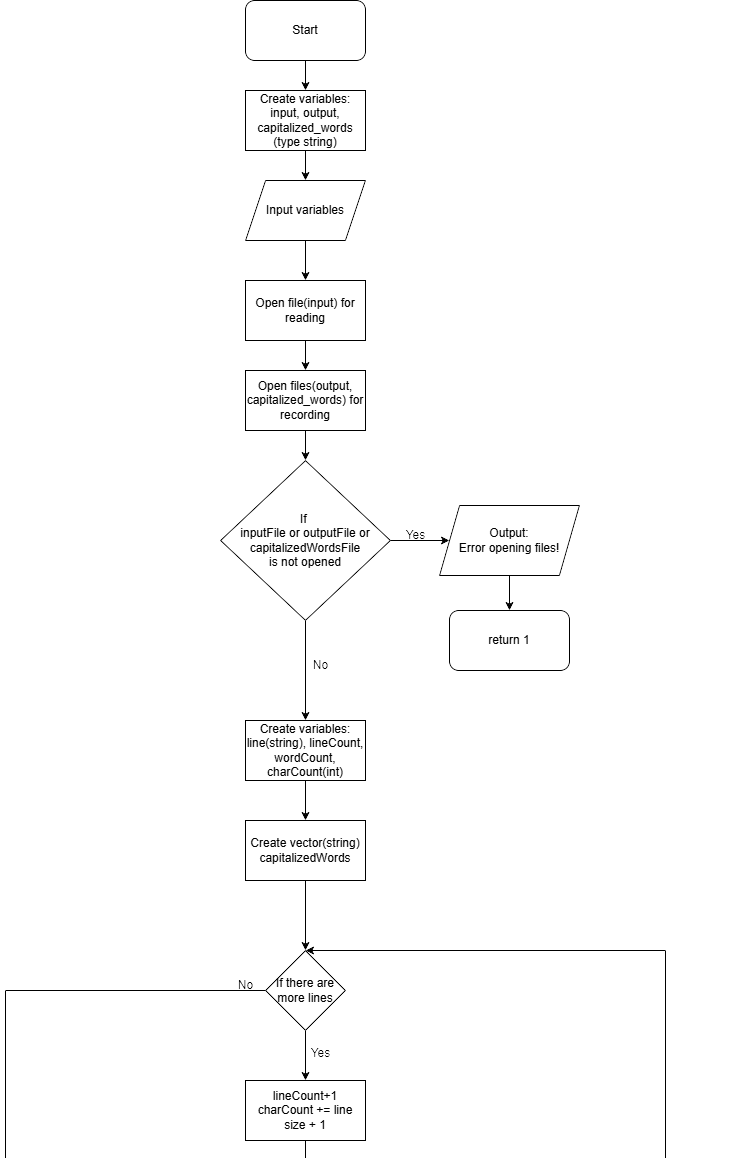


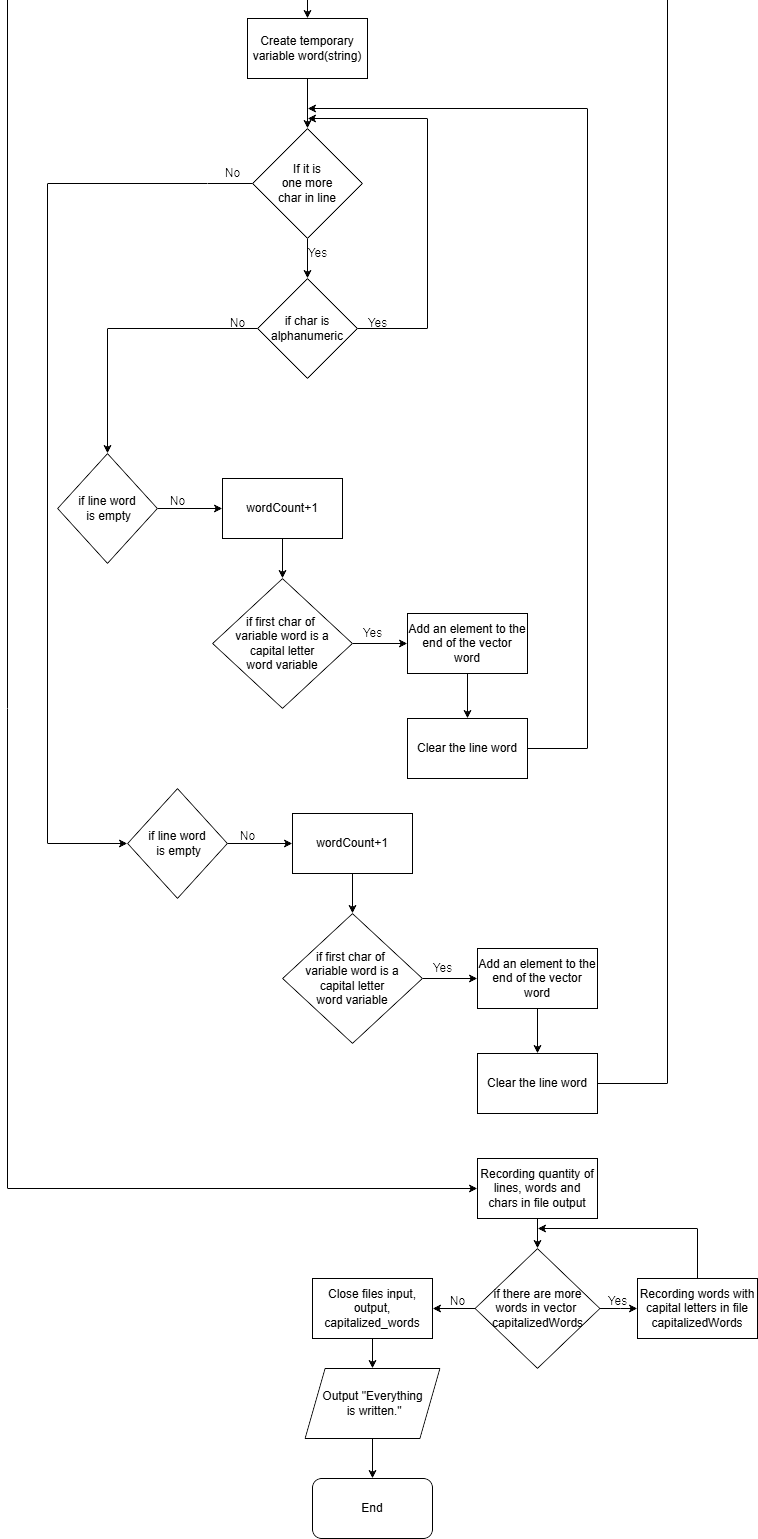
Орієнтовний час виконання: 27хв

Task 5: Algotester practice Lab1v2

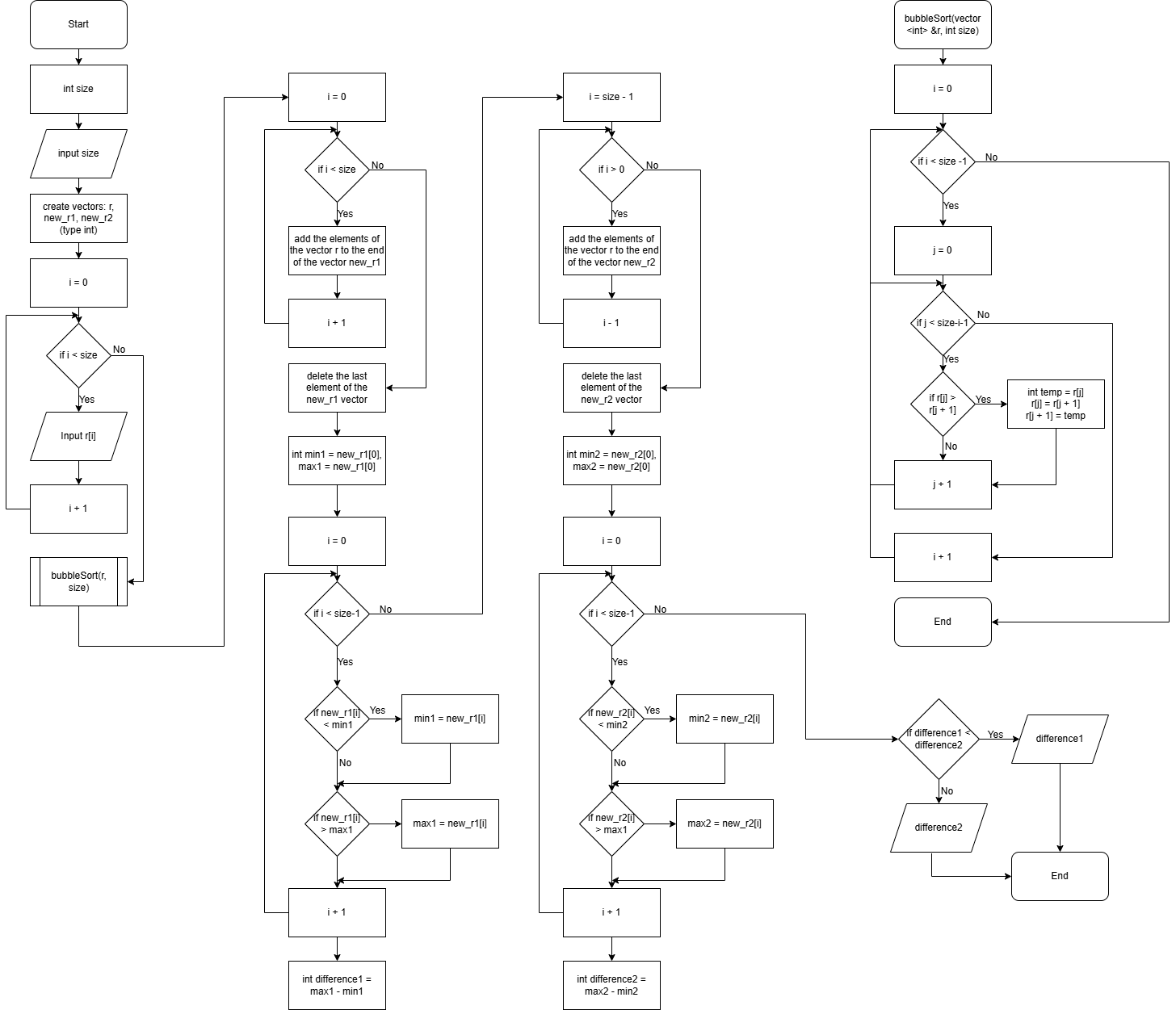


Task 5: Self practice

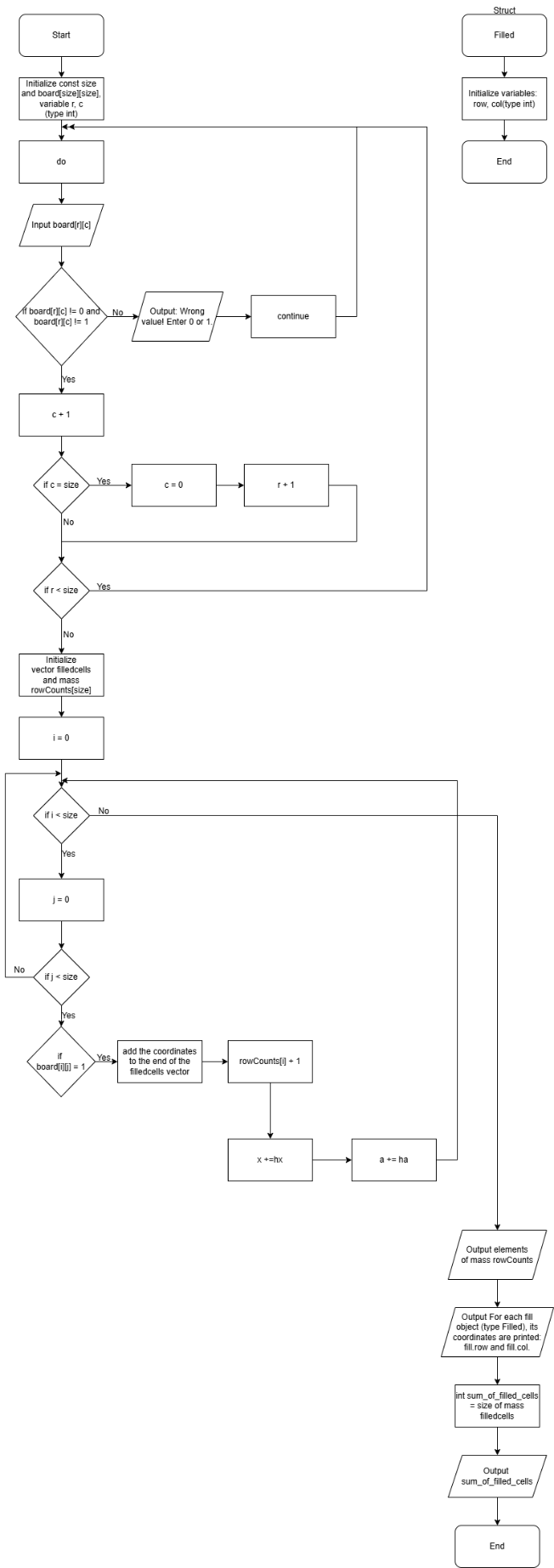




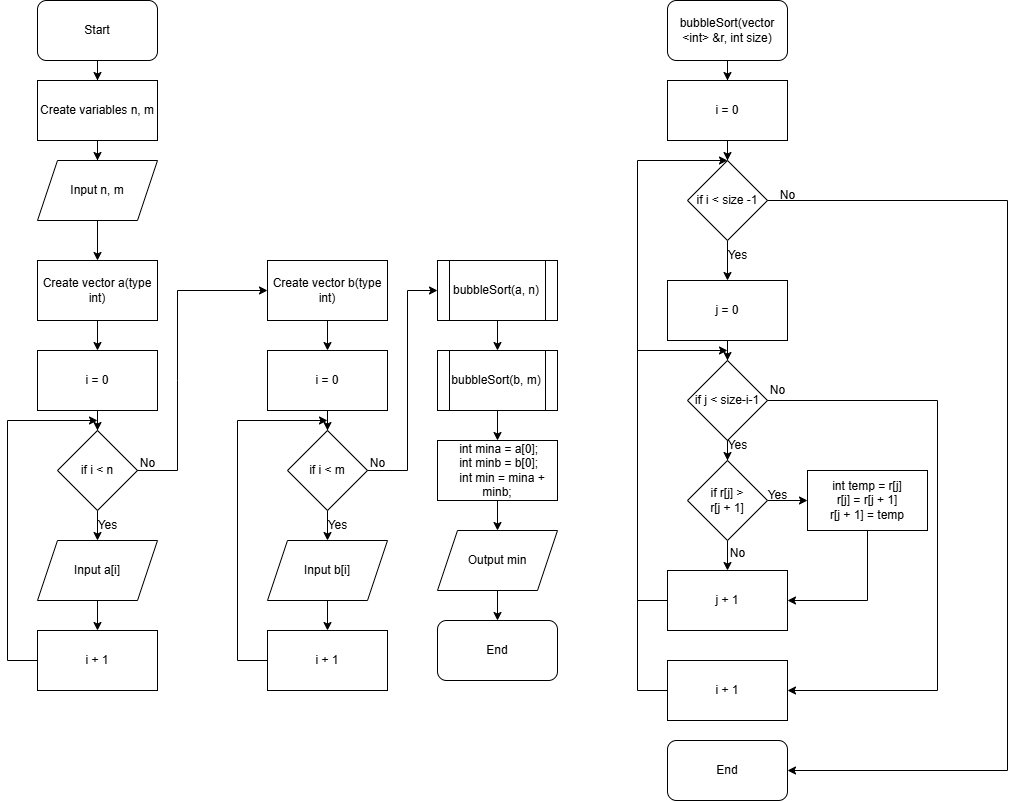
Орієнтовний час виконання: 1 год

Task 6: Algotester practice Lab2v1  


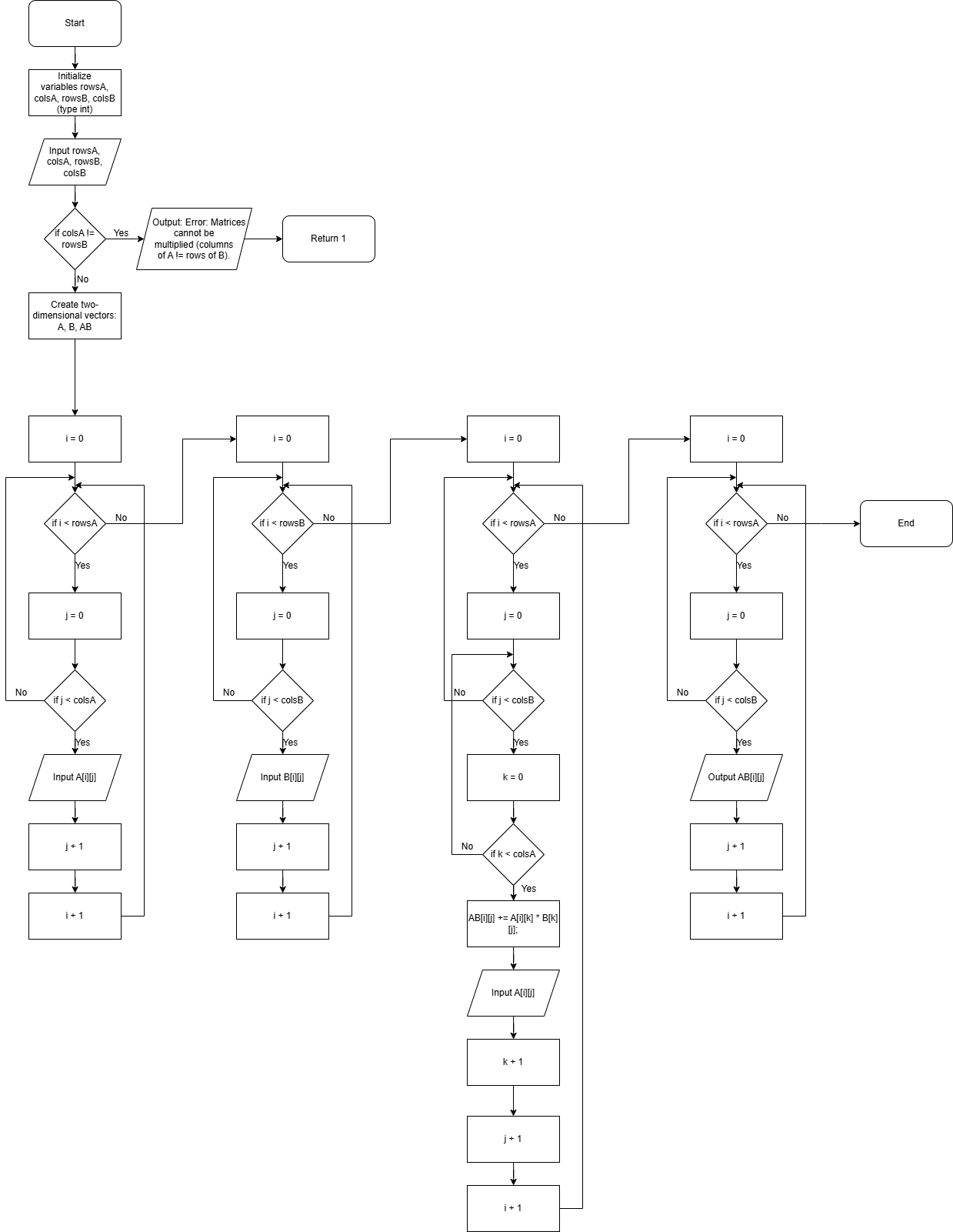
Task 6: Self practice



Орієнтовний час виконання: 40хв

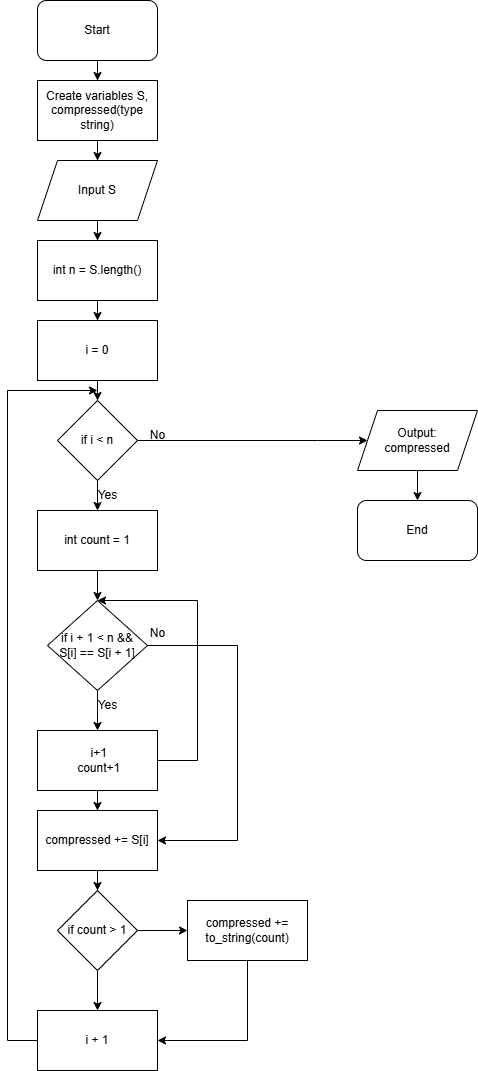
Task 7: Algotester practice Halloween  


Task 7: Self practice

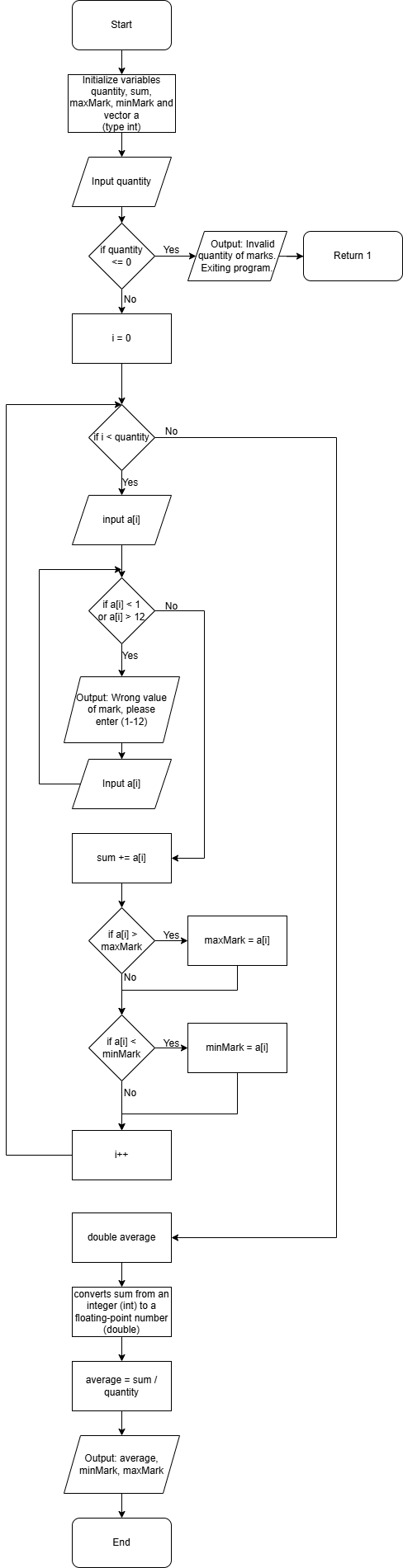


Орієнтовний час виконання: 30хв

Task 8: Algotester practice Lab3v3



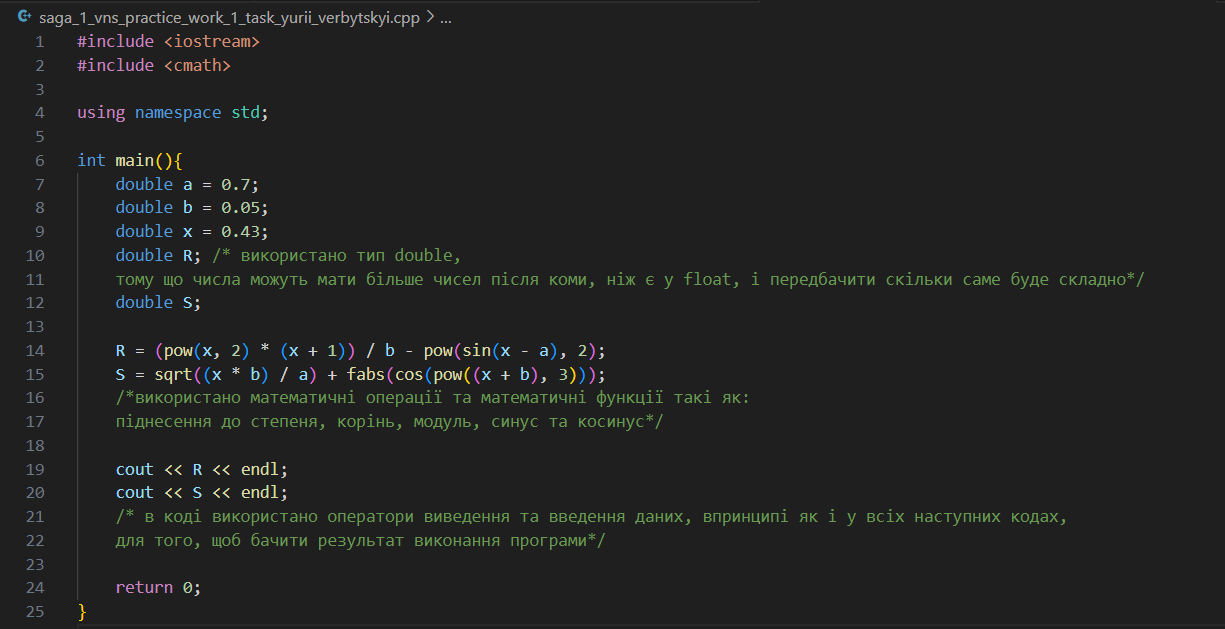
Task 8: Self practice



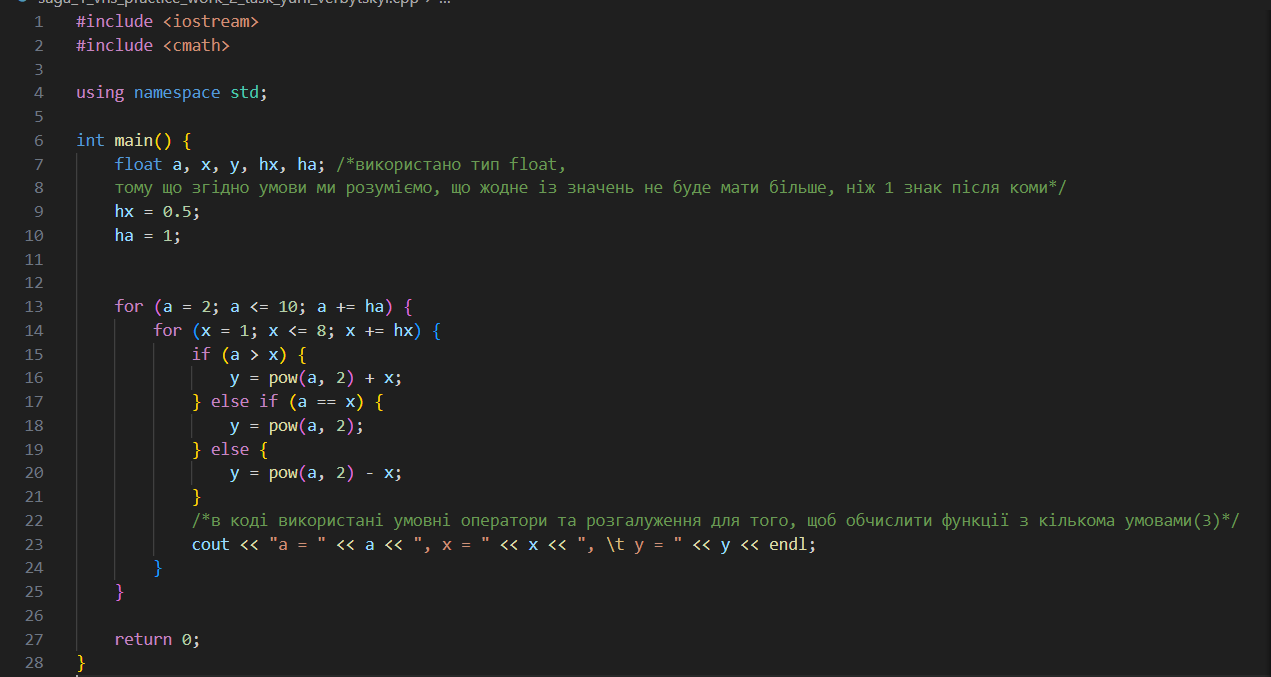
Орієнтовний час виконання: 46хв

**4)Код програм з посиланням на зовнішні ресурси:**

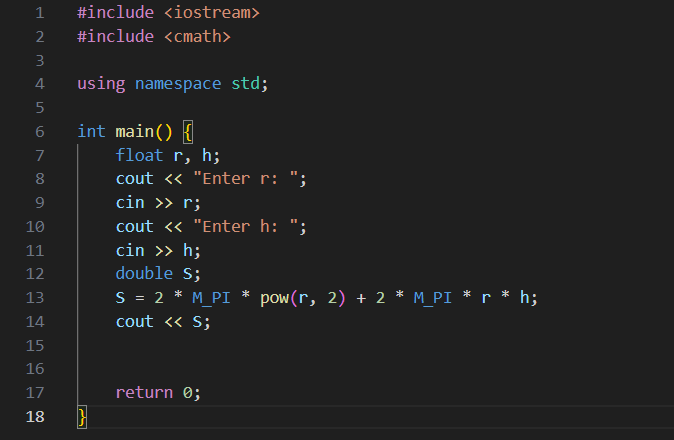
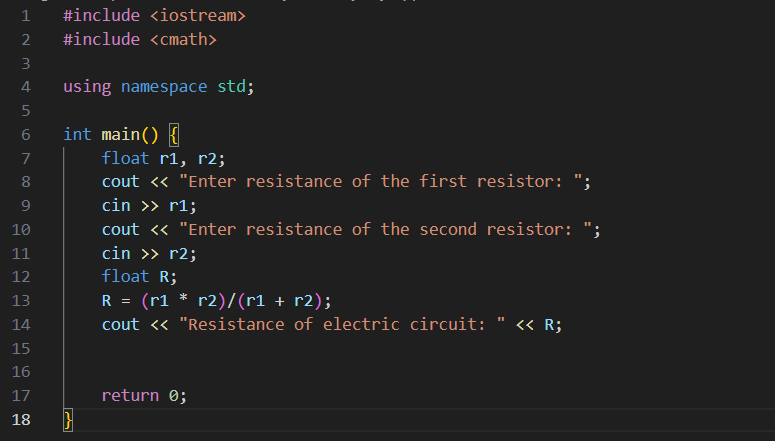
Task 1: VNS practice work 1 task 9



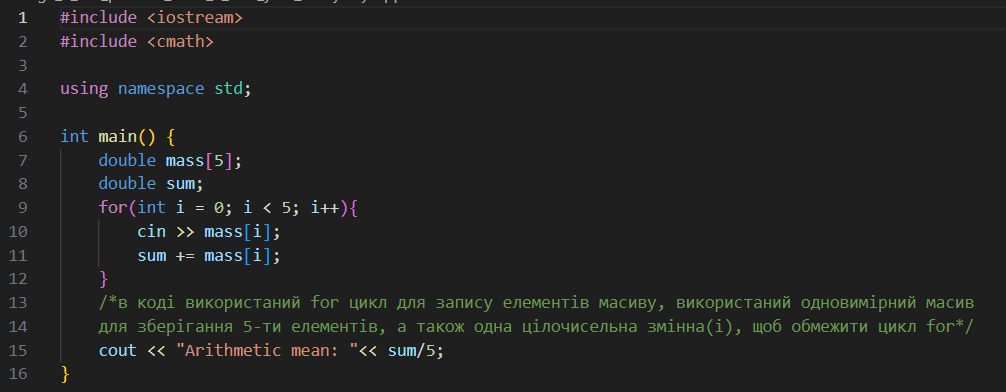
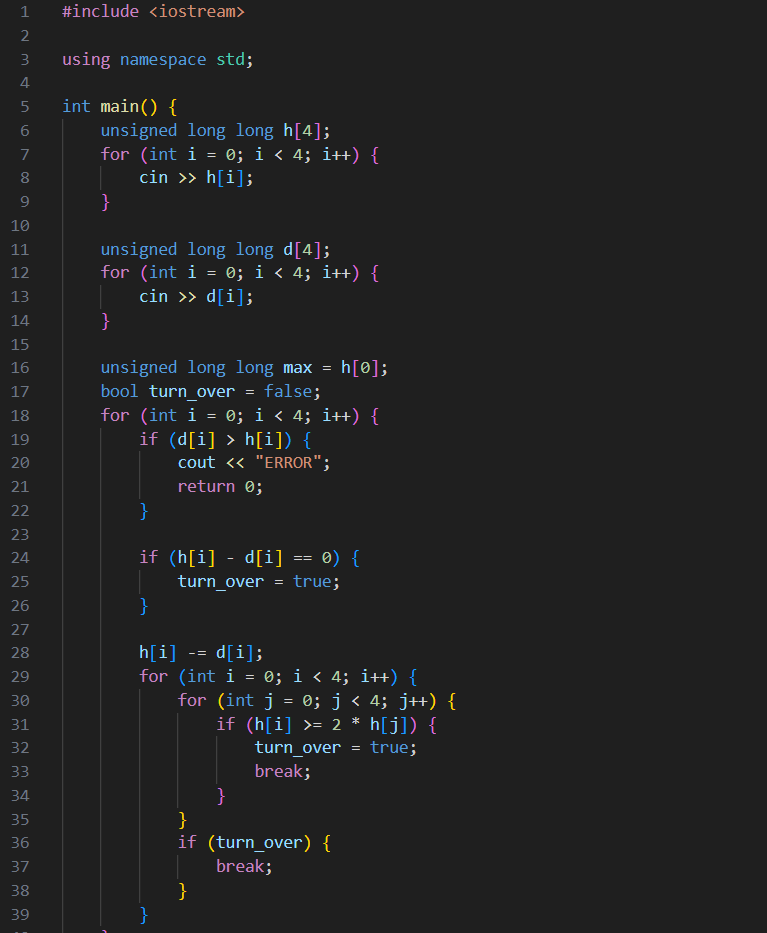
Task 2: VNS practice work 2 task 17

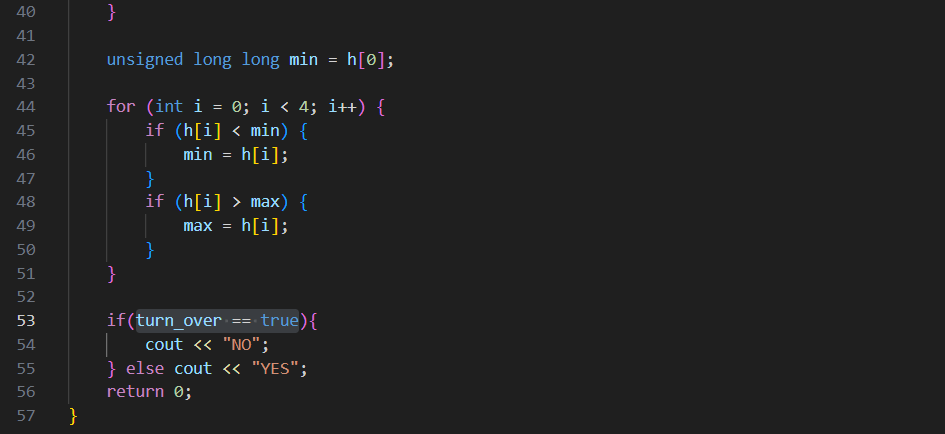


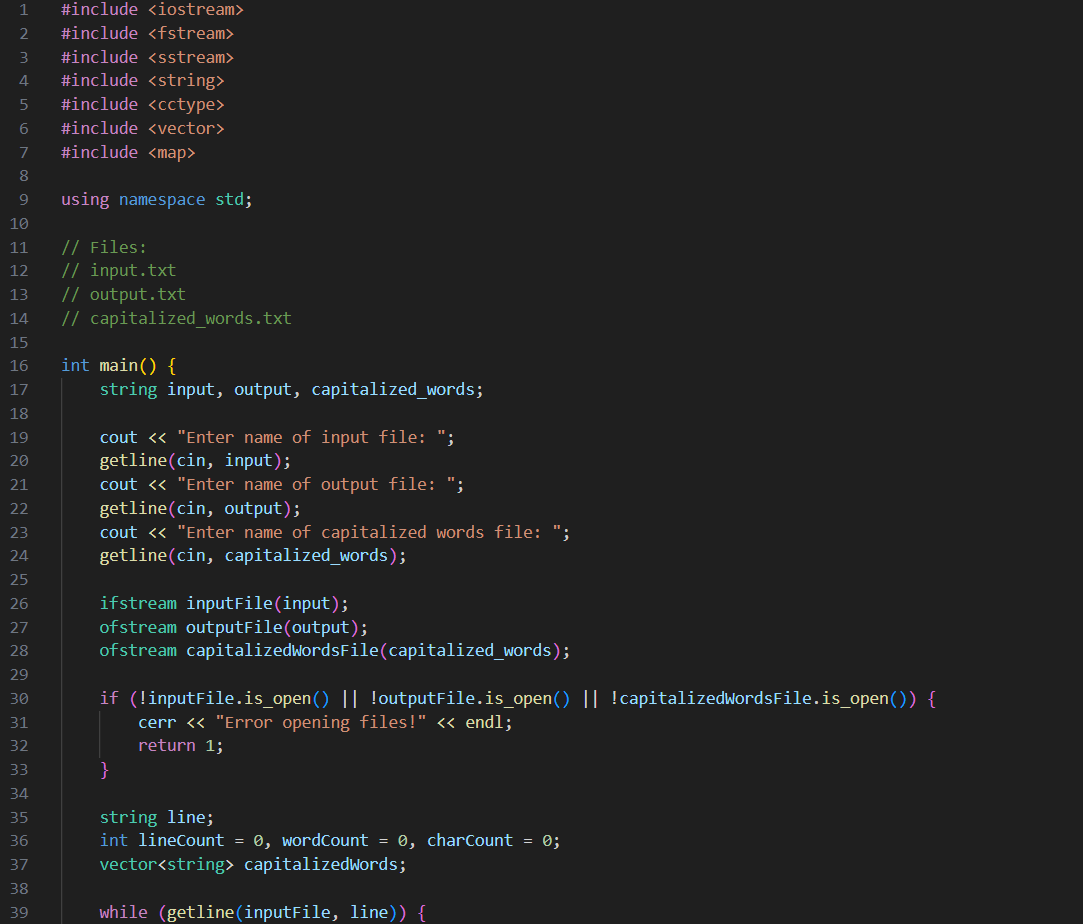
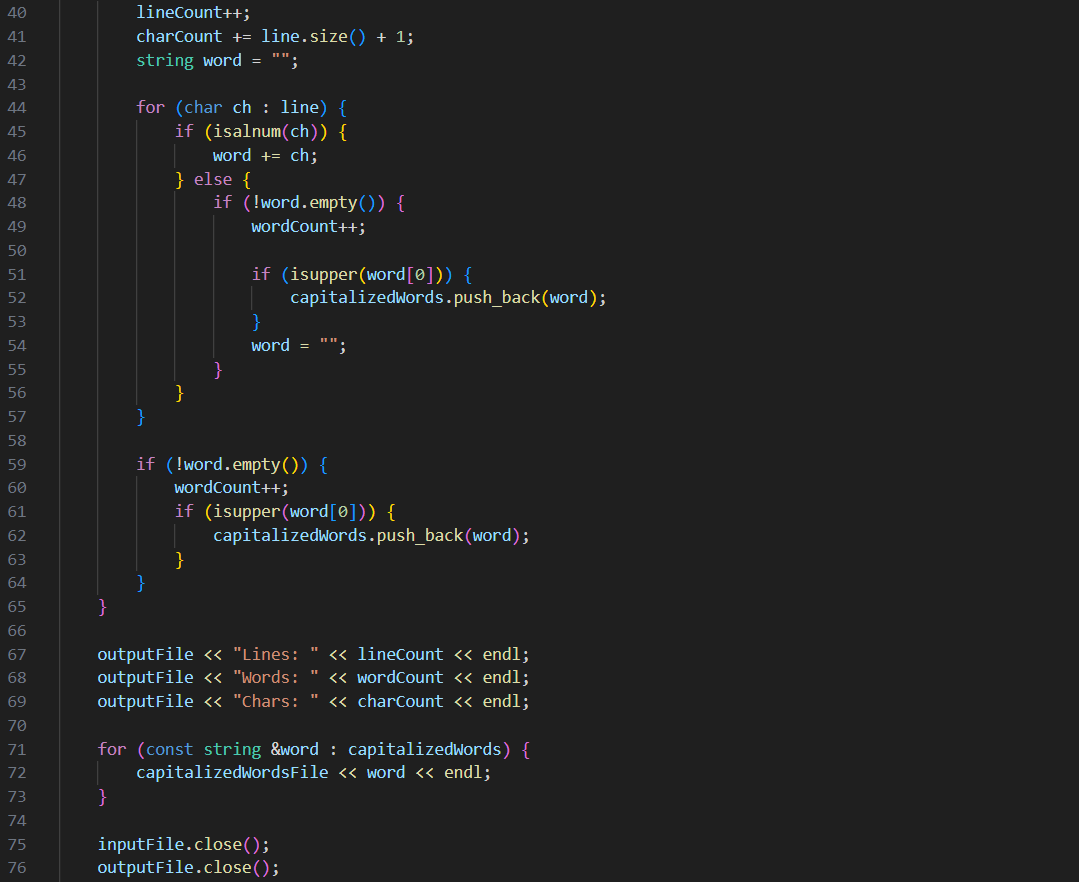
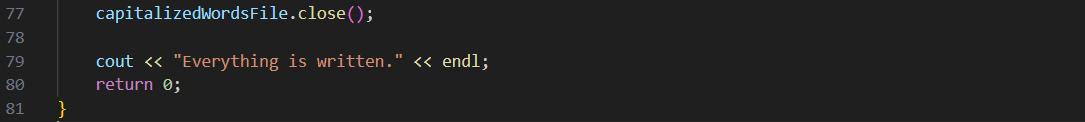
Task 3: VNS practice work 3 task 14

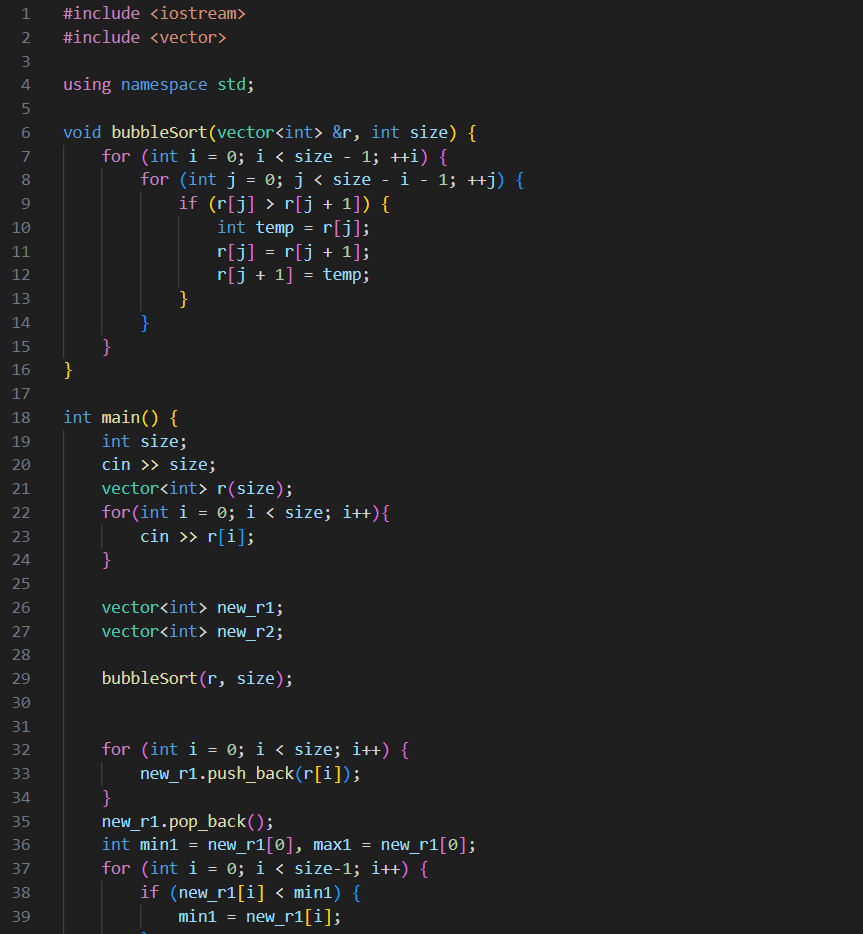
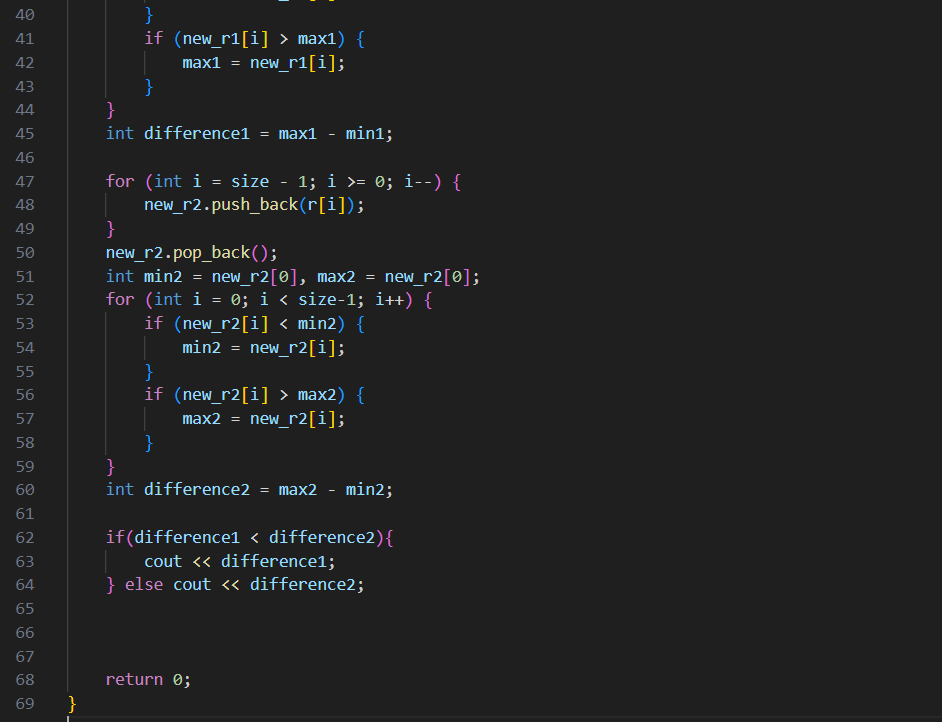
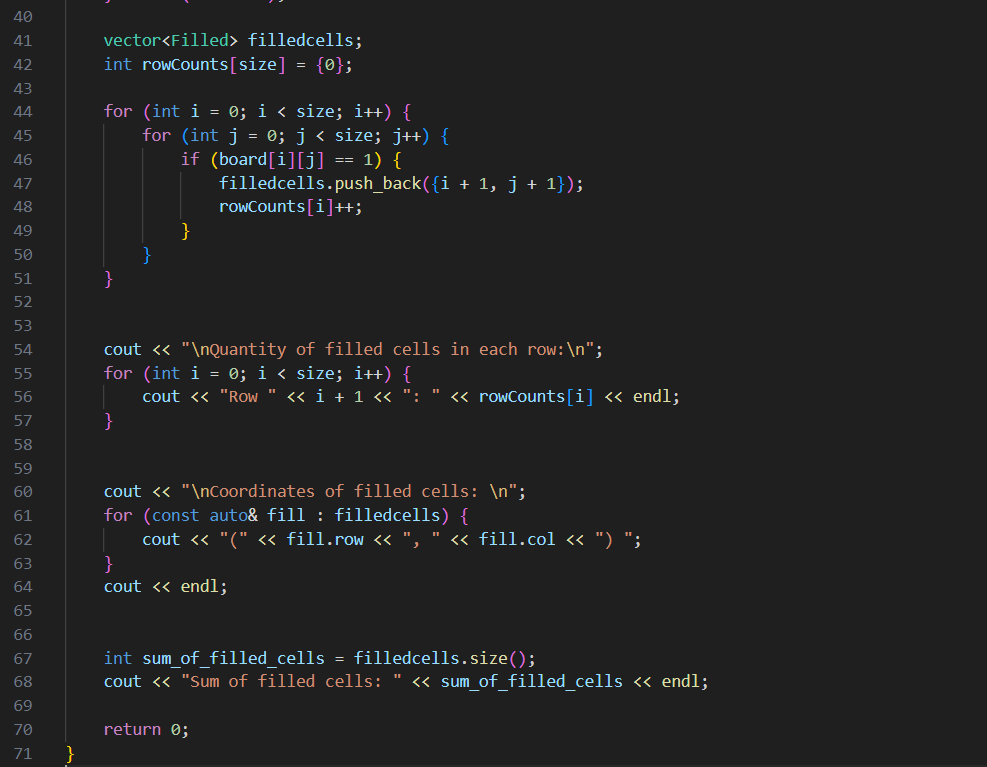
Task 4: VNS practice work 4 task 12

Task 5: Algotester practice work

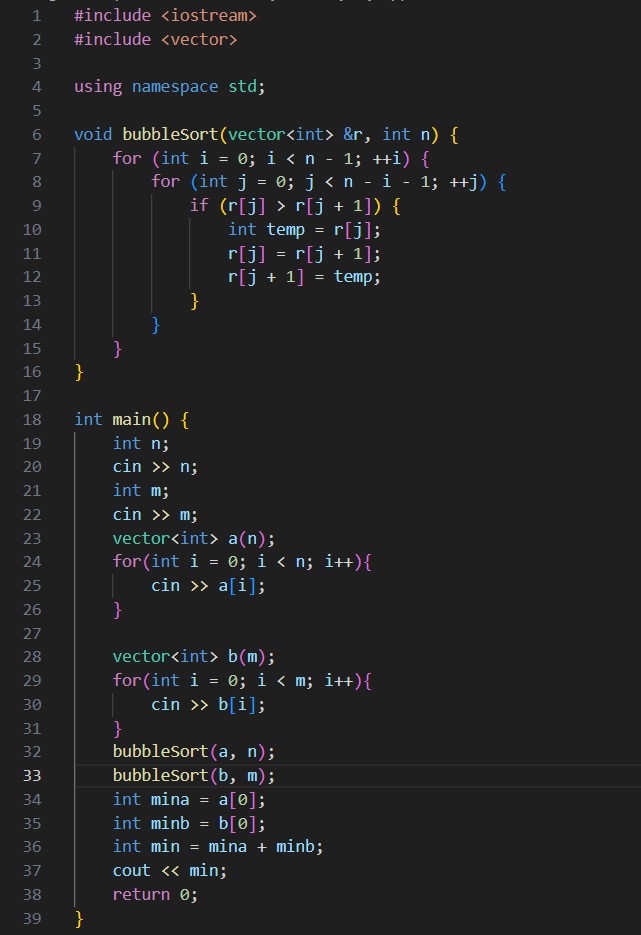


Task 5: Self practice work   
  


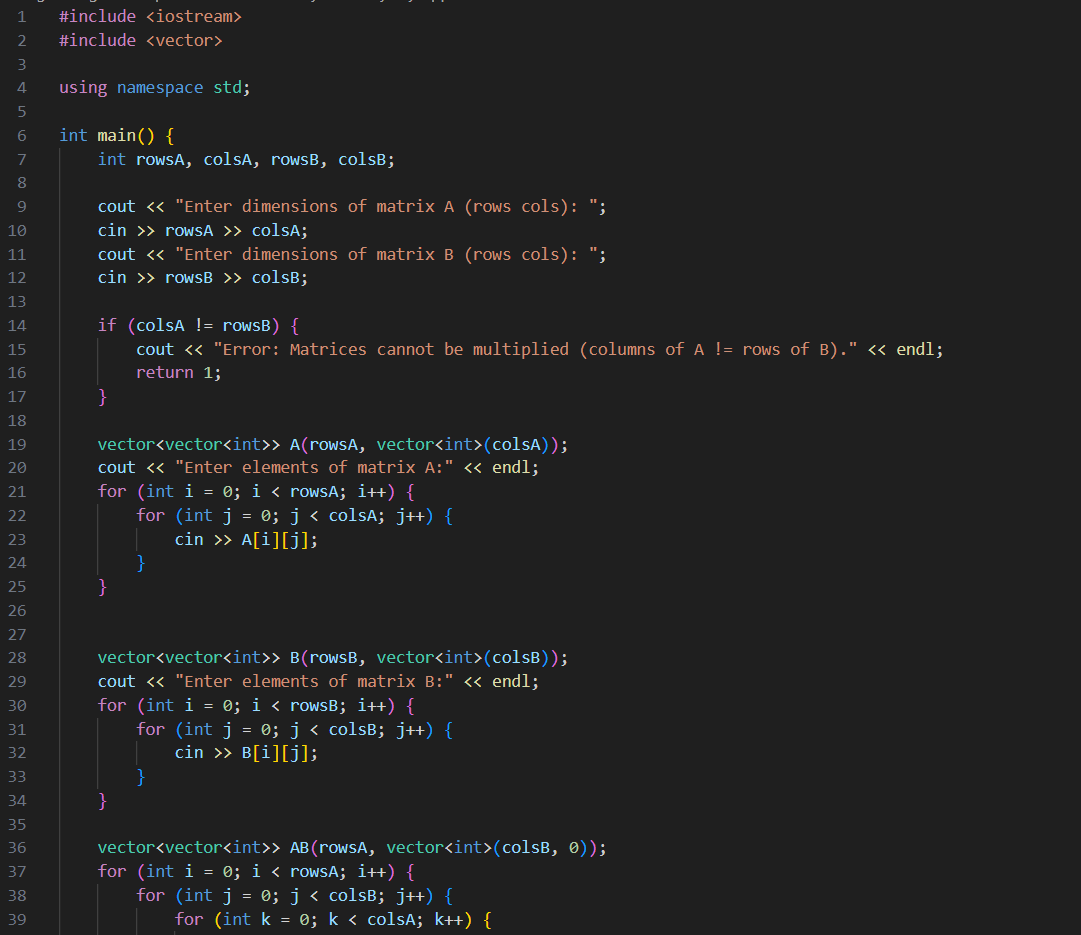
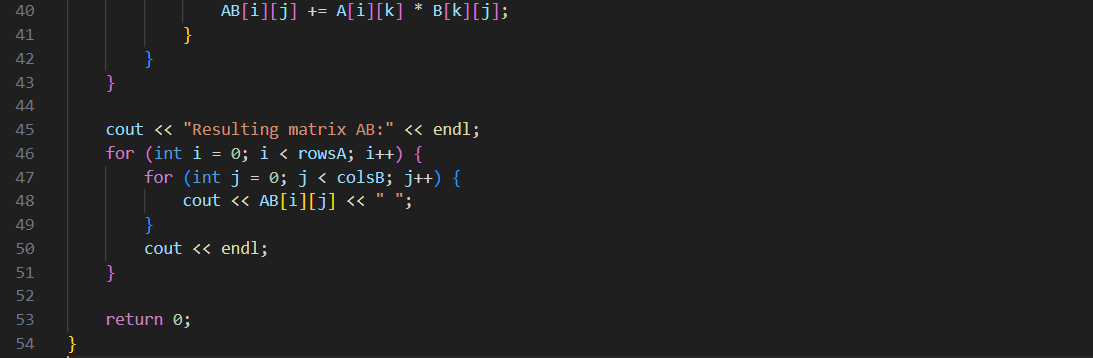
Task 6: Algotester practice work

  
  
Task 6: Self practice work  


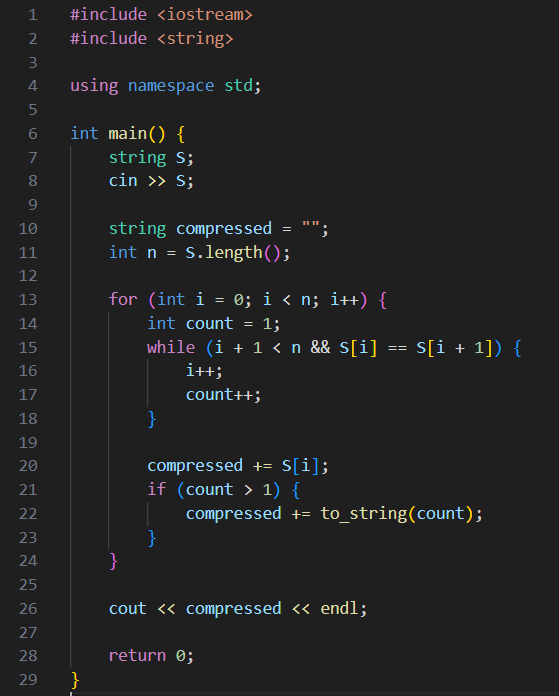
Task 7: Algotester practice work



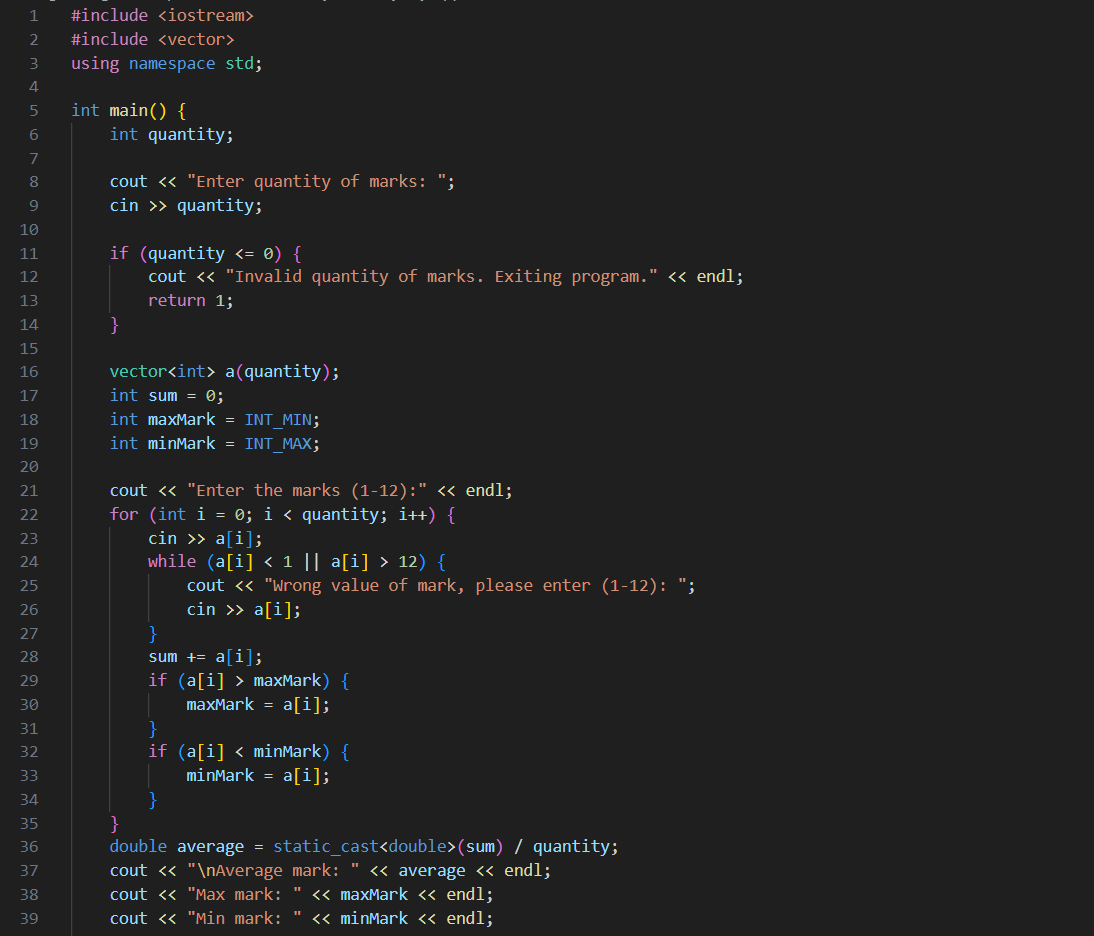
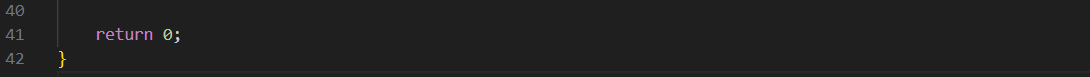
Task 7: Self practice work

Task 8: Algotester practice work



Task 8: Self practice work

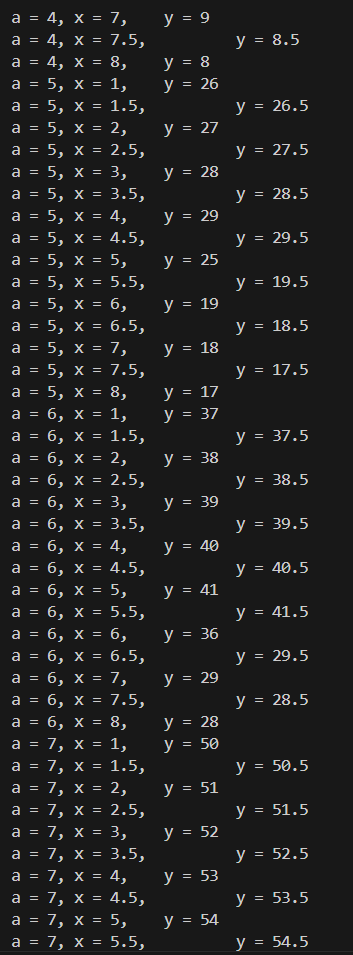
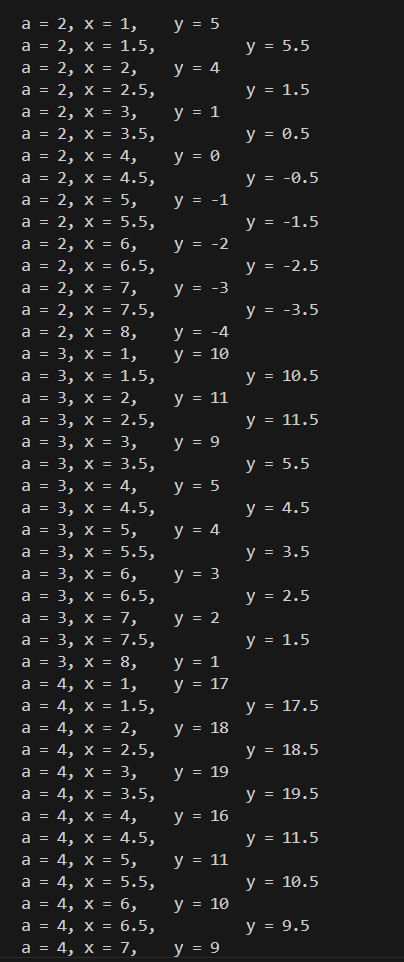
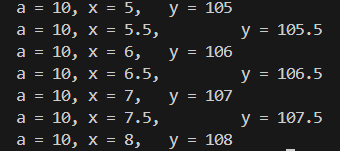
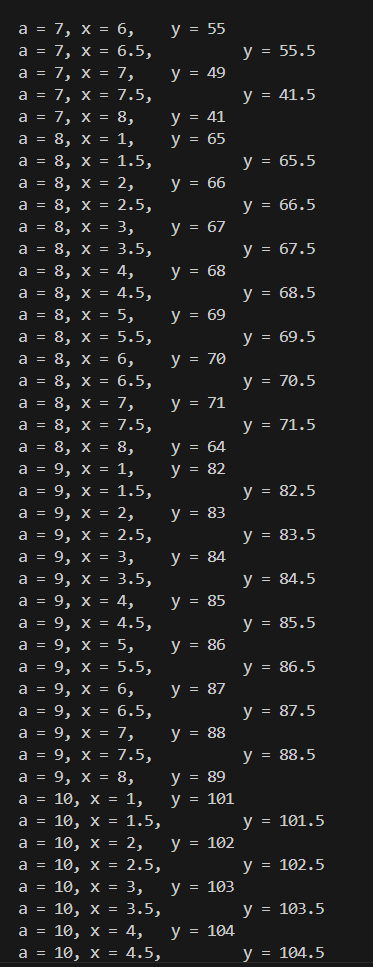
**5) Результати виконання завдань та фактично затрачений час**

Task 1: VNS practice work 1 task 9



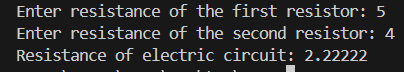
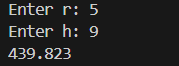
Фактично затрачений час: 7хв

Task 2: VNS practice work 2 task 17

Фактичний час затрачений на виконання: 40хв

Task 3: VNS practice work 3 task 14



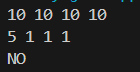
Фактичний час затрачений на виконання: 20хв

Task 4: VNS practice work 4 task 12

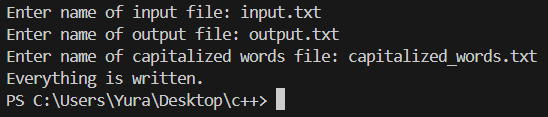
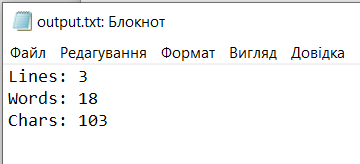
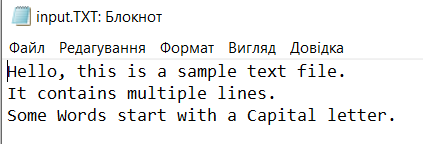
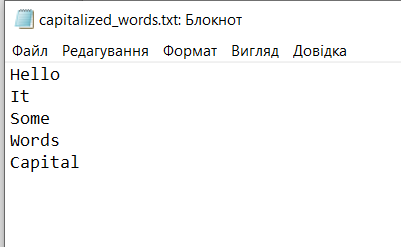


Фактичний час затрачений на виконання: 5хв

Task 5: Algotester practice work

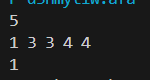


Task 5: Self practice work

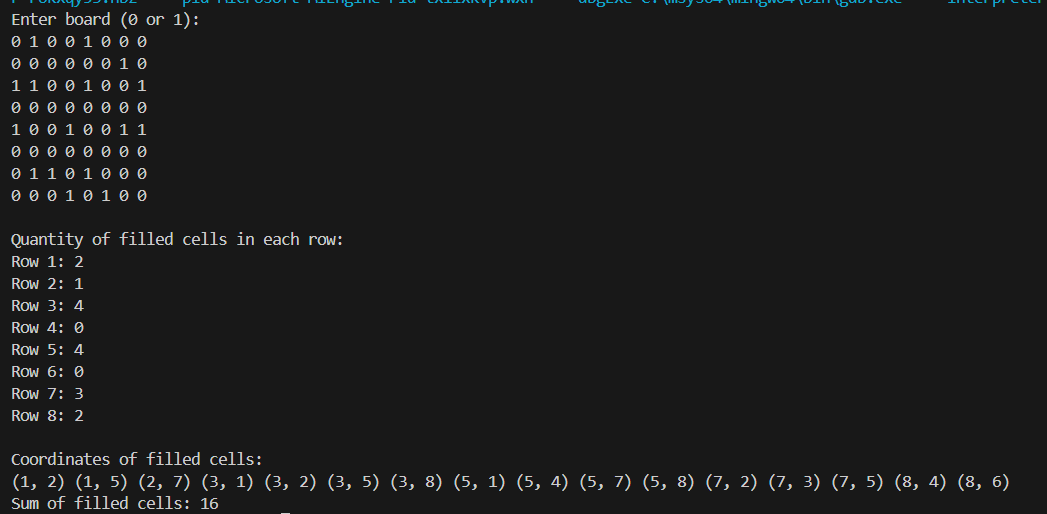
  
  


Фактичний час виконання: 1год 29хв

Task 6: Algotester practice work

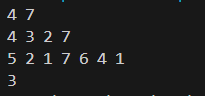


Task 6: Self practice work

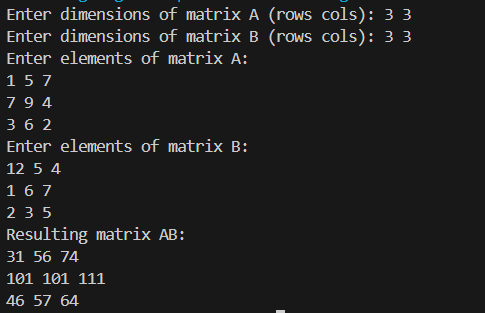


Фактичний час виконання: 25хв

Task 7: Algotester practice work

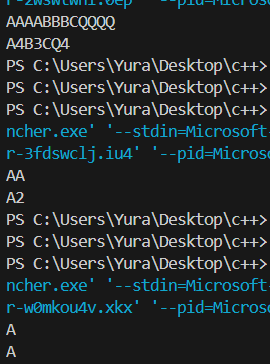


Task 7: Self practice work

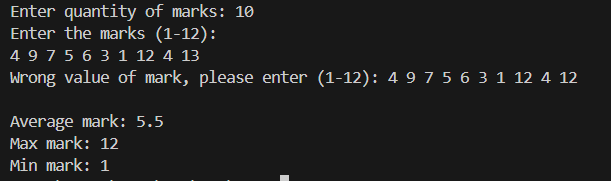


Фактичний час виконання: 40хв

Task 8: Algotester practice work



Task 8: Self practice work



Фактичний час виконання: 27хв

**Висновок:** Опановуючи роботу з масивами різних типів, вказівниками, посиланнями, динамічними структурами даних та алгоритмами їх обробки, я набув 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 структуризованість і логічну цілісність інформації.

**Посилання на пул реквест:** [**https://github.com/artificial-intelligence-department/ai\_programming\_playground\_2024/pull/458**](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/458)