Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

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

***з дисципліни:*** «Мови та парадигми програмування»

до:

ВНС Розрахунково-графічних робіт № 1-4

Практичних Робіт № 7

***Виконав:***

студент групи ШІ-11 Мацько Ілля Феліксович

Львів 2023

# **Тема роботи:**

Розрахункова робота

# **Мета роботи:**

Одержати практичні навички в розробці і дослідженні алгоритмів розв’язання задач.

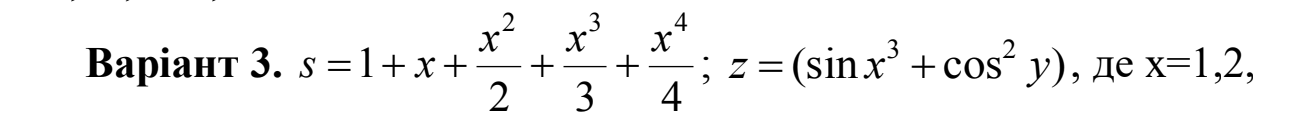
# **Виконання роботи:**

## **Опрацювання завдання та вимог до програм та середовища:**

**Завдання №1 VNS Practice Work - Task 1**

* + Варіант завдання - 3
  + Деталі завдання:

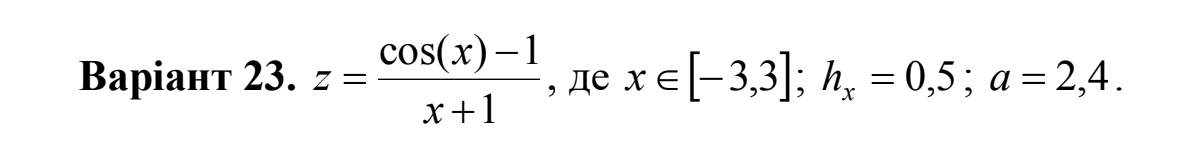
Розробити лінійний алгоритм для розв’язання задачі.



**Завдання №2 VNS Practice Work - Task 2**

* + Варіант завдання - 23
  + Деталі завдання:

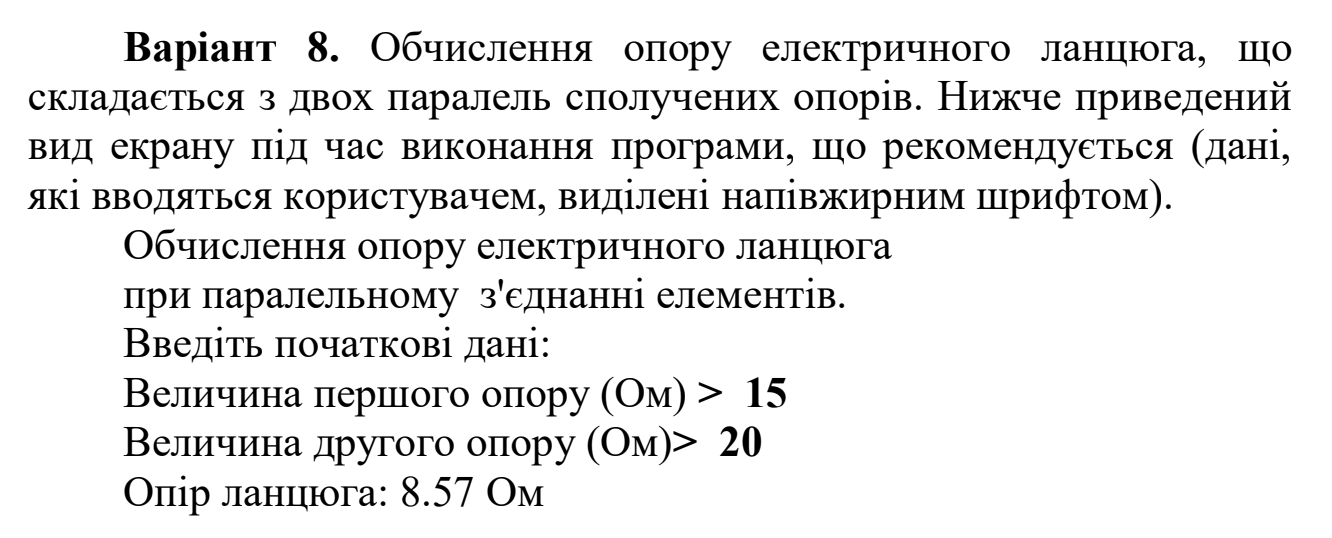
Розробити алгоритм, що розгалужується для розв’язання задачі



**Завдання № 3 VNS Practice Work - Task 3**

* Варіант завдання - 8
* Деталі завдання:

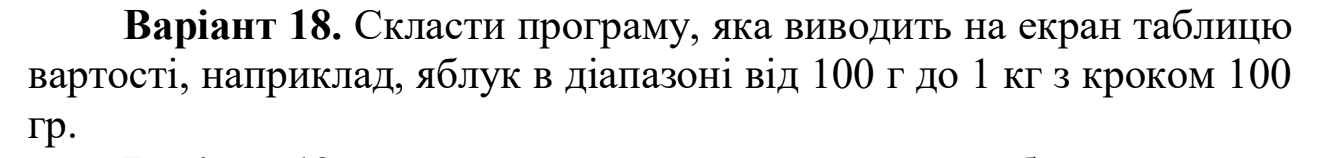
Обчислення опору електричного ланцюга, що складається з двох паралель сполучених опорів.



**Завдання № 4 VNS Practice Work - Task 4**

* Варіант завдання - 18
* Деталі завдання:

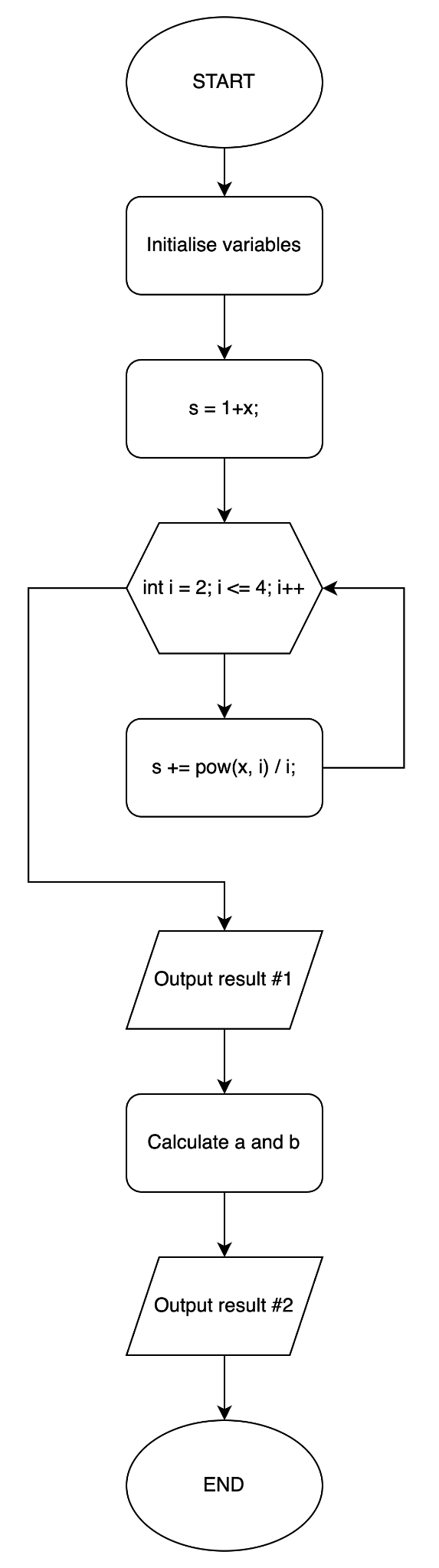
Скласти програму, яка виводить на екран таблицю вартості, наприклад, яблук в діапазоні від 100 г до 1 кг з кроком 100 гр.



## **Дизайн та оцінка часу виконання завдань:**

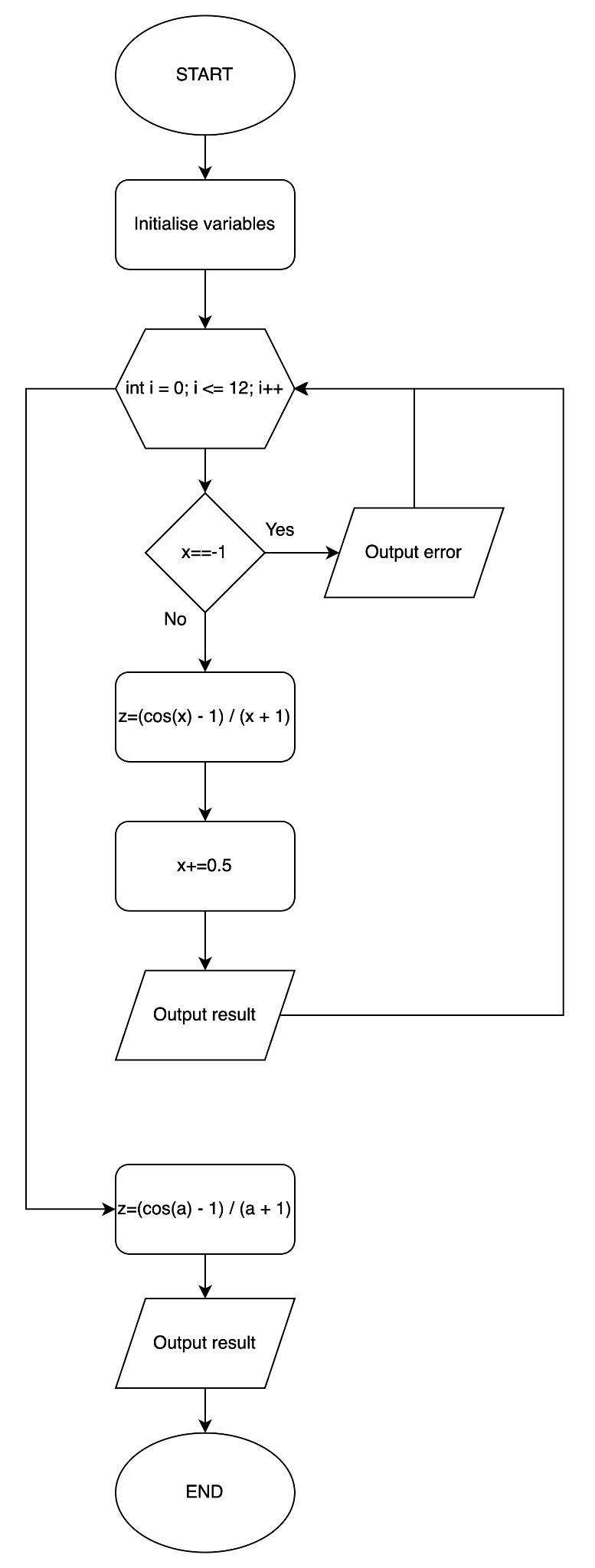
Програма № 1 VNS Practice Work - Task 1

* + Запланований час на реалізацію - 20 хвилин



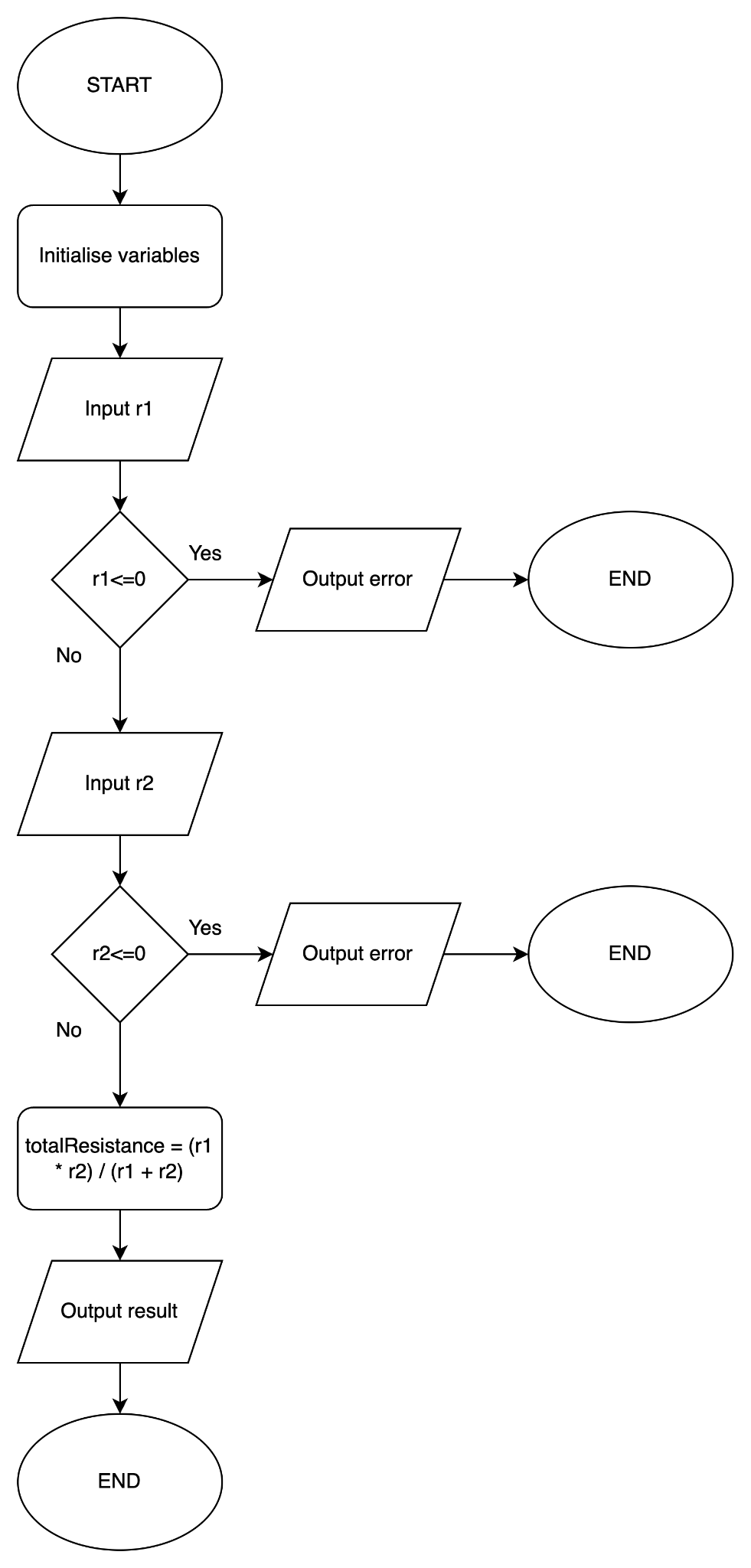
Програма № 2 VNS Practice Work - Task 2

* + Запланований час на реалізацію - 20 хвилин



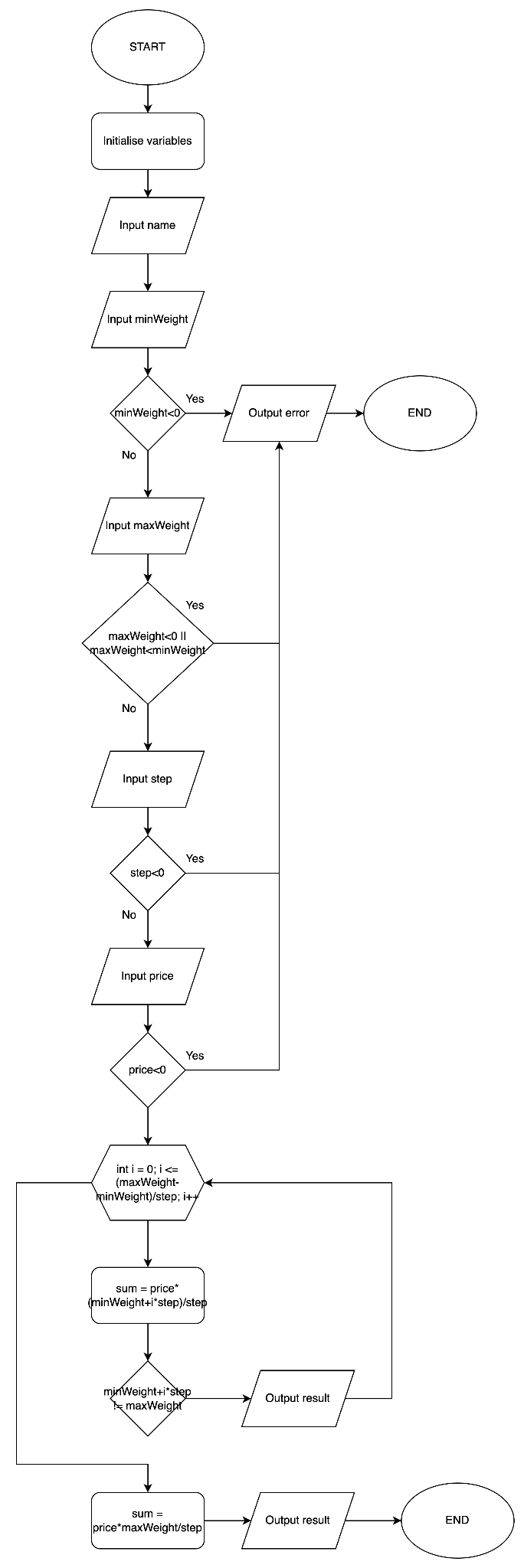
Програма № 3 VNS Practice Work - Task 3

* + Запланований час на реалізацію - 20 хвилин



Програма № 4 VNS Practice Work - Task 4

* + Запланований час на реалізацію - 20 хвилин



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

Завдання № 1 [VNS Practice Work - Task 1](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/788/files#diff-875dad54565b82663b318852e63fa1b60f12255af17842090ff420923ba25e80)

#include <iostream>

#include <cmath> // Включення бібліотеки cmath для функцій pow() та sin()

using namespace std;

int main()

{

// Ініціалізація змінних

double s = 0, z = 0;

double x = 1.2, y = -0.8;

// Розрахунок значення s

s = 1 + x; // Ініціалізація s першим членом

for (int i = 2; i <= 4; i++)

{

s += pow(x, i) / i; // Додавання наступних членів до s за допомогою операцій піднесення до степеня та ділення

}

cout << "s = " << s << endl;

// Розрахунок значень a та b

double a = pow(sin(x), 3); // Розрахунок sin(x) піднятого до степеня 3

double b = pow(cos(y), 2); // Розрахунок cos(y) піднятого до степеня 2

// Розрахунок значення z

z = a + b; // Додавання значень a та b для отримання z

cout << "z = " << z;

}

Завдання № 2 [VNS Practice Work - Task 2](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/788/files#diff-1791518371b8a47a2d313c8387dd2fe959acac539e1444fb8a94212ab87b7512)

#include <iostream>

#include <cmath> // Для використання функції cos()

using namespace std;

//В завданні дані деякі числа, а що це за числа - не написано. Зробив так, як зрозумів.

int main()

{

// Вхідні дані

double x = -3, a = 2.4;

// Обчислення та виведення значень для різних x

for (int i = 0; i <= 12; i++)

{

if (x == -1)

{

cout << "x = " << x << " Error" << endl; // Виводимо помилку, бо не можна ділити на 0

x += 0.5;

continue;

}

// Обчислення та виведення значення виразу для конкретного x

cout << "Using x: x = " << x << " z = " << (cos(x) - 1) / (x + 1) << endl;

x += 0.5;

}

// Виведення результату для конкретного a

cout << endl << "Using a: a = " << a << " z = " << (cos(a) - 1) / (a + 1);

return 0;

}

Завдання № 3 [VNS Practice Work - Task 3](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/788/files#diff-122b52f4b649ed811a3789c1adfe715dff8c4063f47f6587d5af8a94ec8476f0)

#include <iostream>

using namespace std;

int main()

{

// Змінні для опорів

double r1, r2;

// Введення даних

cout << "Обчислення опору електричного ланцюга при\nпаралельному з'єднанні елементів.\n";

cout << "Введіть початкові дані: " << endl;

// Введення значень опорів від користувача

cout << "Величина першого опору (Ом) >> ";

cin >> r1;

// Перевірка на коректність введених даних

if (r1 <= 0)

{

cerr << "Введено некоректну величину опору. Опір повинен бути більше нуля.";

return -1; // Помилка введення

}

cout << "Величина другого опору (Ом) >> ";

cin >> r2;

// Перевірка на коректність введених даних

if (r2 <= 0)

{

cerr << "Введено некоректну величину опору. Опір повинен бути більше нуля.";

return -1; // Помилка введення

}

// Обчислення та виведення результату

double totalResistance = (r1 \* r2) / (r1 + r2);

cout << "Опір ланцюга: " << totalResistance << " Ом";

}

Завдання № 4 [VNS Practice Work - Task 4](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/788/files#diff-5ce09f8c665d2a25e9acb51de3edf3f5356e02502766dd8f1094c00c18d17027)

#include <iostream>

using namespace std;

int main()

{

// Введення даних

cout << "Введіть назву товару: ";

string name;

cin >> name;

cout << "Введіть нижню межу ваги(у грамах): ";

int minWeight;

cin >> minWeight;

if(minWeight < 0)

{

cout << "Некоректні введені дані. Будь ласка, перевірте введені значення.";

return -1; // Помилка введення

}

cout << "Введіть верхню межу ваги(у грамах): ";

int maxWeight;

cin >> maxWeight;

if(maxWeight < 0 || maxWeight < minWeight)

{

cout << "Некоректні введені дані. Будь ласка, перевірте введені значення.";

return -1; // Помилка введення

}

cout << "Введіть крок ваги(у грамах): ";

double step;

cin >> step;

if(step <= 0)

{

cout << "Некоректні введені дані. Будь ласка, перевірте введені значення.";

return -1; // Помилка введення

}

cout << "Введіть ціну за " << step << " г: ";

int price;

cin >> price;

if (price <= 0)

{

cout << "Некоректні введені дані. Будь ласка, перевірте введені значення.";

return -1; // Помилка введення

}

for(int i = 0; i <= (maxWeight-minWeight)/step; i++)

{

double sum = price\*(minWeight+i\*step)/step; // Розрахунок суми

if(minWeight+i\*step != maxWeight)

cout << "Вага: " << minWeight+i\*step << " г, Ціна: " << sum << " грн" << endl; // Виведення результату

}

// Розрахунок для максимальної ваги

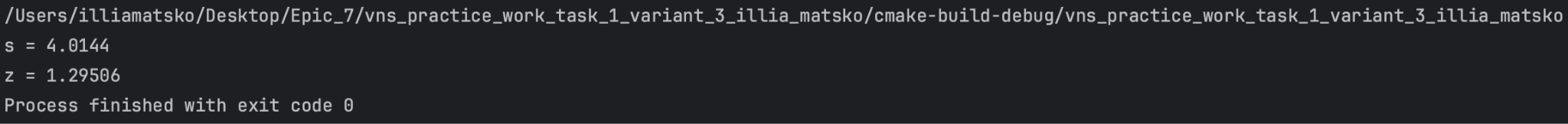
double sum = price\*maxWeight/step;

cout << "Вага: " << maxWeight << " г, Ціна: " << sum << " грн";

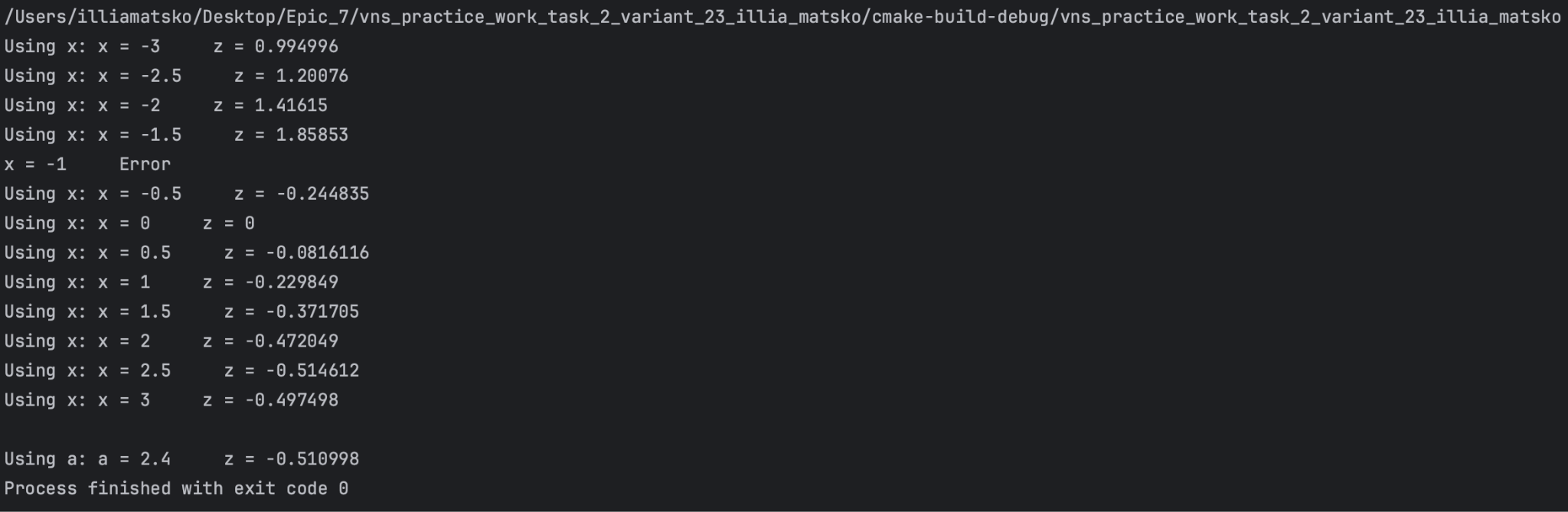
}

1. **Результати виконання завдань, тестування та фактично витрачений час:**

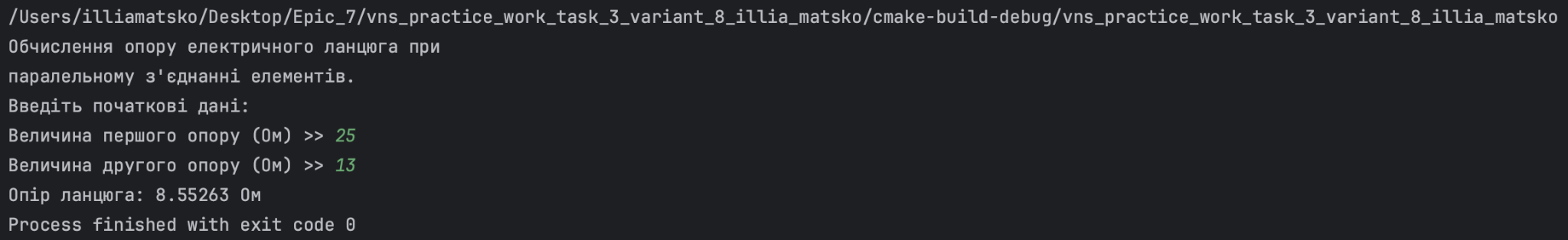
**Завдання № 1**



**Завдання № 2**

****

**Завдання № 3**

****

**Завдання № 4**

****

[**Посилання на Pull-request**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/788)

**Контрольні запитання**

1. Основні властивості алгоритму:

Властивості алгоритму: дискретність, визначеність, виконуваність, скінченність, результативність, масовість, ефективність.

Дискретність (латинською discretus — розділений, розривний) алгоритму означає, що виконання алгоритму зводиться до виконання окремих дій (кроків) у певній послідовності. Причому, кожну вказівку алгоритму виконують за скінченний проміжок часу.

Визначеність (однозначність) означає, що алгоритм однозначно визначає порядок дій виконавця, результат цих дій і не потребує додаткового тлумачення. Алгоритм не може містити вказівки, які виконавець може сприймати неоднозначно.

Виконуваність - це означає, що алгоритм, призначений для певного виконавця, може містити лише вказівки, які входять до системи вказівок цього виконавця.

Скінченність означає, що виконання алгоритму закінчиться після скінченної (можливо, досить великої) кількості кроків і за скінченний час для довільних вхідних даних.

Результативність - це означає, що після закінчення виконання алгоритму обов’язково:

* або одержують розв'язок-результат відповідно до поставленої мети;
* або встановлюють відсутність таких розв'язків (множина розв'язків порожня);
* або встановлюють неможливість розв'язання задачі розвэі.

Масовість алгоритму означає, що алгоритм можна застосувати до цілого класу однотипних задач, для яких спільними є умова та хід розв’язування та які відрізняються лише початковими (вхідними) даними.

Ефективність алгоритму описує час виконання і об'єм ресурсів, необхідних для виконання алгоритму: чим менше часу (часова ефективність) і ресурсів (просторова ефективність), тим ефективність вища.

2. Алгоритм:

Алгоритм — це запис скінченої послідовності вказівок, виконання яких призводить до розв'язання певної задачі.

3. Основні етапи розробки алгоритмів:

- Формалізація задачі

- Розробка алгоритму

- Виконання алгоритму та аналіз результатів

- Виправлення та оптимізація алгоритму, якщо необхідно

4. Базові конструкції:

- Послідовність

- Розгалуження

- Цикл

5. Складні базові конструкції:

- Підпрограми (функції, процедури)

- Масиви та структури даних

6. Конструкція розгалуження:

- Конструкція розгалуження - це вибір між двома або більше шляхами виконання програми в залежності від умови.

7. Конструкція циклу:

- Конструкція циклу - це повторення виконання певного фрагмента коду доти, доки виконується певна умова.

8. Правило виконання циклу з передумовою:

- Цикл з передумовою виконується, доки вказана умова істинна. Умова перевіряється перед кожним виконанням тіла циклу.

9. Правило виконання циклу з відомою кількістю повторень:

- Цикл з відомою кількістю повторень виконується певну кількість разів, визначену заздалегідь.

10. Обчислювальна складність алгоритму:

- Обчислювальна складність алгоритму визначається кількістю ресурсів (часу, пам'яті), які потрібно для виконання алгоритму.

11. Оцінка обчислювальної складності:

- Велике "O" (O-велике) використовується для визначення асимптотичної верхньої межі часової і просторової складності алгоритму.

12. Рекурсивні функції:

- Рекурсивні функції - це функції, які викликають самих себе. Переваги використання рекурсії включають зручність, ясність та здатність до розв'язання проблеми за допомогою меншої кількості коду.

# **Висновки:**

В ході виконання роботи надавались практичні навички розробки та вивчення алгоритмів. Застосовано базові конструкції алгоритмів у мові програмування C++, вирішено конкретні завдання. Вивчено принципи введення та виведення даних, а також використано конструкції розгалуження та циклу. Робота сприяла закріпленню теоретичних знань та розвитку навичок практичного програмування, що має важливе значення для подальшого вивчення комп'ютерних наук**.**