

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



Звіт

про виконання лабораторних та практичних робіт блоку № 3

На тему: «Цикли. Вкладені Цикли. Завершення виконання циклів.
Функції. Простір імен. Перевантаження функцій. Функції з змінною
кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції.»

з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи № 2

ВНС Лабораторної Роботи № 3

ВНС Лабораторної Роботи № 7

Практичних Робіт до блоку № 3

Виконав(ла):

Тема роботи:

Цикли. Вкладені цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції з змінною кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції.

Мета роботи:

Дослідження та впровадження різних типів циклів, включаючи прості та вкладені цикли, для автоматизації повторюваних дій у програмах. Це забезпечило глибше розуміння структур керування потоком виконання програм, а також способів оптимізації циклічних процесів.

Вивчення механізмів завершення виконання циклів, зокрема команд break та continue, що дозволило контролювати роботу циклів і керувати виходом з них за певних умов.

Опанування функцій як основного інструменту для розбиття програми на логічні модулі, зокрема робота з передачею аргументів та поверненням значень. Вивчення простору імен для уникнення конфліктів між ідентифікаторами і поліпшення читабельності та структури коду.

Перевантаження функцій, що дозволило створювати функції з однаковими іменами, але різними типами або кількістю параметрів, для підвищення гнучкості і багаторазовості коду.

Опанування функцій з змінною кількістю параметрів (еліпсис), що сприяло написанню універсальних функцій, здатних обробляти різну кількість аргументів, що підвищило адаптивність програми до різних сценаріїв використання.

Робота з рекурсією для вирішення складних завдань

Застосування вбудованих функцій для виконання типових операцій, таких як математичні розрахунки або робота з рядками, що підвищило ефективність розробки та зменшило кількість ручного коду.

Теоретичні відомості:

У даній роботі розглядаються основні концепції програмування, зокрема цикли та вкладені цикли для організації повторюваних дій, а також механізми завершення їх виконання. Особливу увагу приділено функціям, їх перевантаженню та використанню простору імен для організації коду. Досліджено функції з змінною кількістю параметрів (еліпсис) і рекурсію, що дозволяють реалізовувати більш гнучкі та складні алгоритми. Також розглянуто вбудовані функції, що спрощують виконання стандартних операцій.

Джерела :

книга - Stephen Prata - “ *C++ Primer Plus* ”

книга - Aditya Y.Bhargava - “ *Grokking algorithms* ”

Завдання № 3

Requirements :

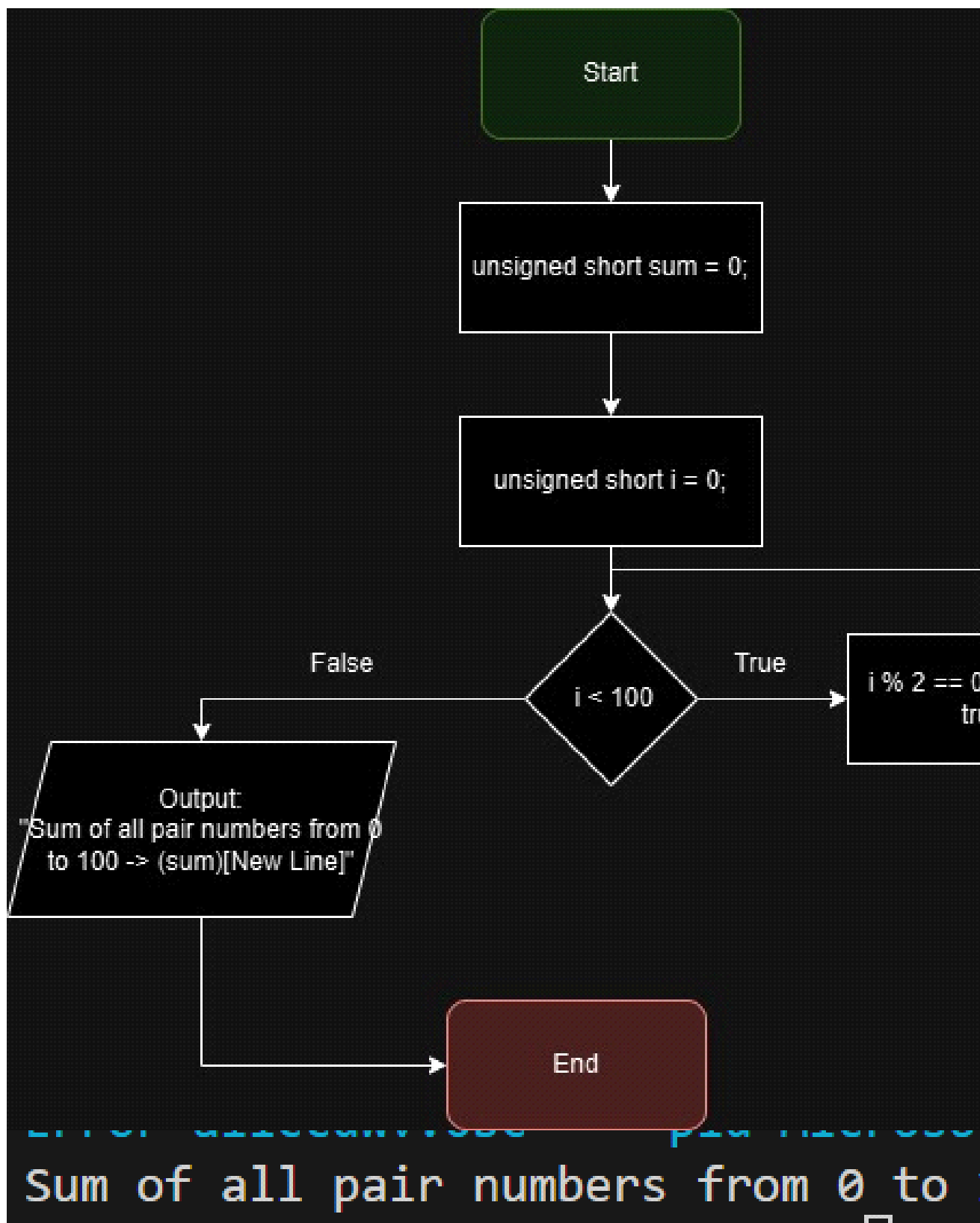
VNS Lab 2

Time:

Expected: 15 min

Spent: 15 min

```
1  #include<iostream>
2
3  int main()
4  {
5      unsigned short sum = 0;
6      for (unsigned short i = 0; i < 100; i++)
7      {
8          i % 2 == 0 ? sum++ : true;
9      }
10     std::cout << "Sum of all pair numbers from 0 to 100 -> " << sum;
11     return 0;
12 }
```



Завдання № 4

Requirements :

VNS Lab 3

Time:

Expected: 1h

Spent: ~1h

```

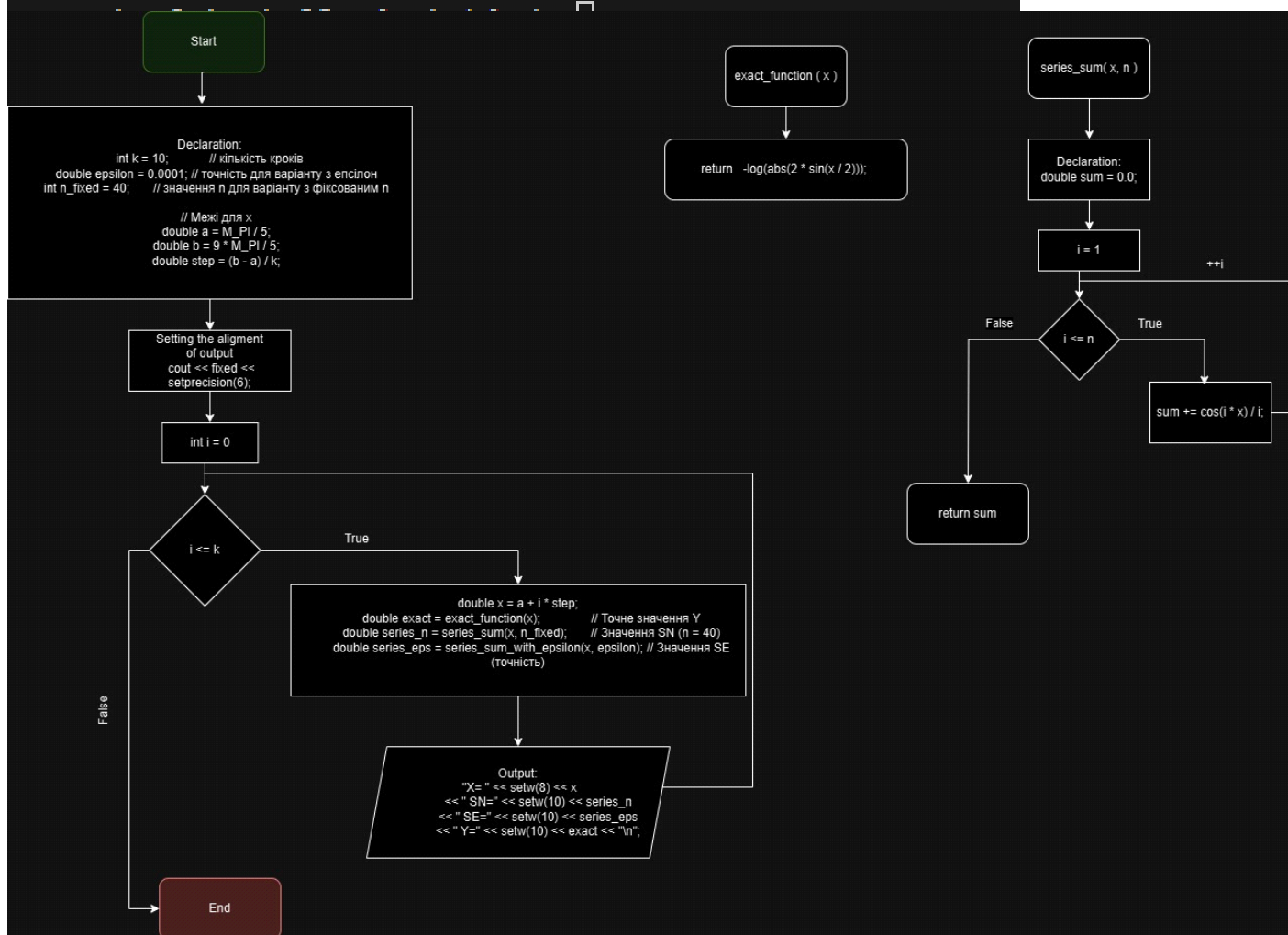
1  #define _USE_MATH_DEFINES
2  #include <iostream>
3  #include <cmath>
4  #include <iomanip>
5
6  using namespace std;
7
8  // Функція для обчислення точного значення f(x)
9  double exact_function(double x) {
10     return -log(abs(2 * sin(x / 2)));
11 }
12
13 // Функція для обчислення степеневого ряду до n-го члена
14 double series_sum(double x, int n) {
15     double sum = 0.0;
16     for (int i = 1; i <= n; ++i) {
17         sum += cos(i * x) / i;
18     }
19     return sum;
20 }
21
22 // Функція для обчислення ряду з заданою точністю
23 double series_sum_with_epsilon(double x, double epsilon) {
24     double sum = 0.0;
25     double term = 0.0;
26     int i = 1;
27     do {
28         term = cos(i * x) / i;
29         sum += term;
30         ++i;
31     } while (abs(term) > epsilon);
32     return sum;
33 }
34
35 int main() {
36     int k = 10; // кількість кроків
37     double epsilon = 0.0001; // точність для варіанту з епсілон
38     int n_fixed = 40; // значення n для варіанту з фіксованим n
39
40     // Межі для x
41     double a = M_PI / 5;
42     double b = 9 * M_PI / 5;
43     double step = (b - a) / k;
44
45     // Обчислення для різних x
46     cout << fixed << setprecision(6);
47     for (int i = 0; i <= k; ++i) {
48         double x = a + i * step;
49         double exact = exact_function(x); // Точне значення Y
50         double series_n = series_sum(x, n_fixed); // Значення SN (n = 40)
51         double series_eps = series_sum_with_epsilon(x, epsilon); // Значення SE (точність)
52
53         cout << "X= " << setw(8) << x
54             << " SN=" << setw(10) << series_n
55             << " SE=" << setw(10) << series_eps
56             << " Y=" << setw(10) << exact << "\n";
57     }
58
59     return 0;
60 }

```

```

X= 0.628319 SN= 0.492090 SE= 0.481735 Y= 0.481212
X= 1.130973 SN= -0.046795 SE= -0.070397 Y= -0.069203
X= 1.633628 SN= -0.380002 SE= -0.377793 Y= -0.377023
X= 2.136283 SN= -0.575094 SE= -0.560627 Y= -0.561108
X= 2.638938 SN= -0.660466 SE= -0.661380 Y= -0.661226
X= 3.141593 SN= -0.680803 SE= -0.693097 Y= -0.693147
X= 3.644247 SN= -0.660466 SE= -0.661380 Y= -0.661226
X= 4.146902 SN= -0.575094 SE= -0.560627 Y= -0.561108
X= 4.649557 SN= -0.380002 SE= -0.377793 Y= -0.377023
X= 5.152212 SN= -0.046795 SE= -0.070397 Y= -0.069203
X= 5.654867 SN= 0.492090 SE= 0.481735 Y= 0.481212

```



Завдання № 5

Requirements :

VNS Lab 7

Time:

Expected: 20m

Spent: 20m

```
#include<iostream>
#include<cstdarg>

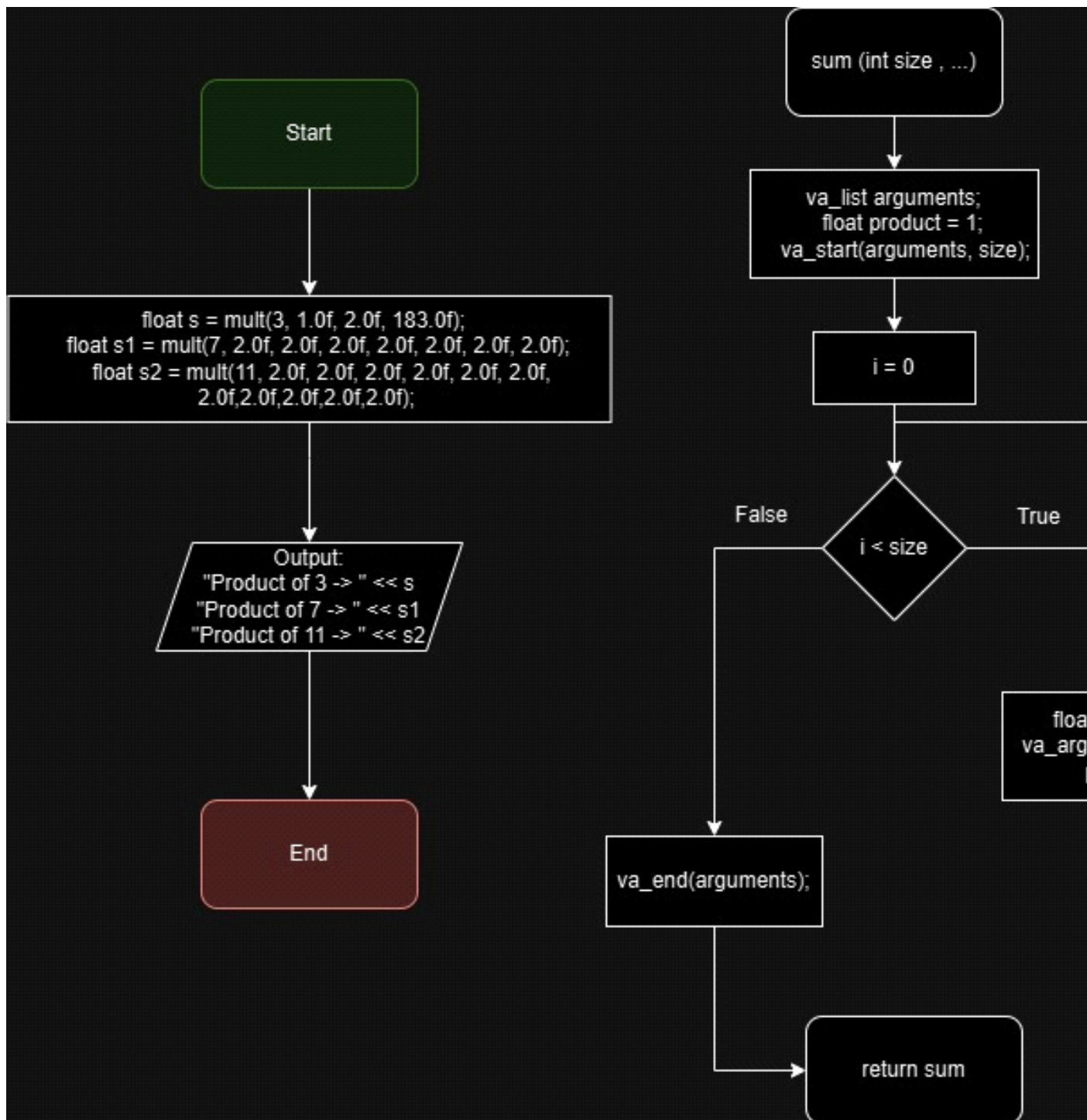
float mult(int size, ...)
{
    va_list arguments;
    float product = 1;
    va_start(arguments, size);
    for (unsigned short i = 0; i < size; i++)
    {
        float element = (float) va_arg(arguments, double);
        product *= element;
    }
    va_end(arguments);
    return product;
}

int main(void)
{
    float s = mult(3, 1.0f, 2.0f, 183.0f);
    float s1 = mult(7, 2.0f, 2.0f, 2.0f, 2.0f, 2.0f, 2.0f, 2.0f);
    float s2 = mult(11, 2.0f, 2.0f, 2.0f, 2.0f, 2.0f, 2.0f, 2.0f, 2.0f, 2.0f, 2.0f);
    std::cout << "Product of 3 -> " << s << std::endl;
    std::cout << "Product of 7 twos -> " << s1 << std::endl;
    std::cout << "Product of 11 twos -> " << s2 << std::endl;
    return 0;
}
```

Product of 3 -> 366

Product of 7 twos -> 128

Product of 11 twos -> 2048



Завдання № 6

Requirements :

Class Practise Task

Time:

Expected: 20m

Spent: 20m

```

#include <iostream>
#include <vector>
#include <string>

using namespace std;

int main() {
    vector<string> books = { "Book1", "Book2", "Book3", "Book4" };
    vector<bool> availability = { true, true, true, true };

    int option;

mainMenu:
    cout << "\nМеню:\n";
    cout << "1. Перерахувати всі книги\n";
    cout << "2. Взяти книгу\n";
    cout << "3. Повернути книгу\n";
    cout << "4. Вийти\n";
    cout << "Оберіть опцію: ";
    cin >> option;

    switch (option) {
    case 1: {
        cout << "Список книг:\n";
        for (size_t i = 0; i < books.size(); i++) {
            cout << (i + 1) << ". " << books[i] << (availability[i] ? " (Доступна)" : " (Не доступна)") << endl;
        }
        break;
    }
    case 2: {
        int choice;
        cout << "Введіть номер книги, яку хочете взяти: ";
        cin >> choice;

        if (choice < 1 || choice > books.size() || !availability[choice - 1]) {
            cout << "Неправильний вибір або книга недоступна.\n";
            goto mainMenu;
        }

        availability[choice - 1] = false;
        cout << "Ви позичили '" << books[choice - 1] << "'\n";
        break;
    }
    case 3: {
        int choice;
        cout << "Введіть номер книги, яку хочете повернути: ";
        cin >> choice;

        if (choice < 1 || choice > books.size() || availability[choice - 1]) {
            cout << "Неправильний вибір або книга не була позичена.\n";
            goto mainMenu;
        }

        availability[choice - 1] = true;
        cout << "Ви повернули '" << books[choice - 1] << "'\n";
        break;
    }
    case 4:
        cout << "Вихід...\n";
        return 0;
    default:
        cout << "Неправильний вибір, спробуйте ще раз.\n";
        goto mainMenu;
    }

    cout << "Бажаєте виконати іншу операцію? (1 - Так, 0 - Ні): ";
    int anotherOperation;
    cin >> anotherOperation;

    if (anotherOperation == 1) {
        goto mainMenu;
    }

    return 0;
}

```

Меню:

1. Перерахувати всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти

Оберіть опцію: 1

Список книг:

1. Book1 (Доступна)
2. Book2 (Доступна)
3. Book3 (Доступна)
4. Book4 (Доступна)

Бажаєте виконати іншу операцію? (1 - Так, 0 - Ні): 1

Меню:

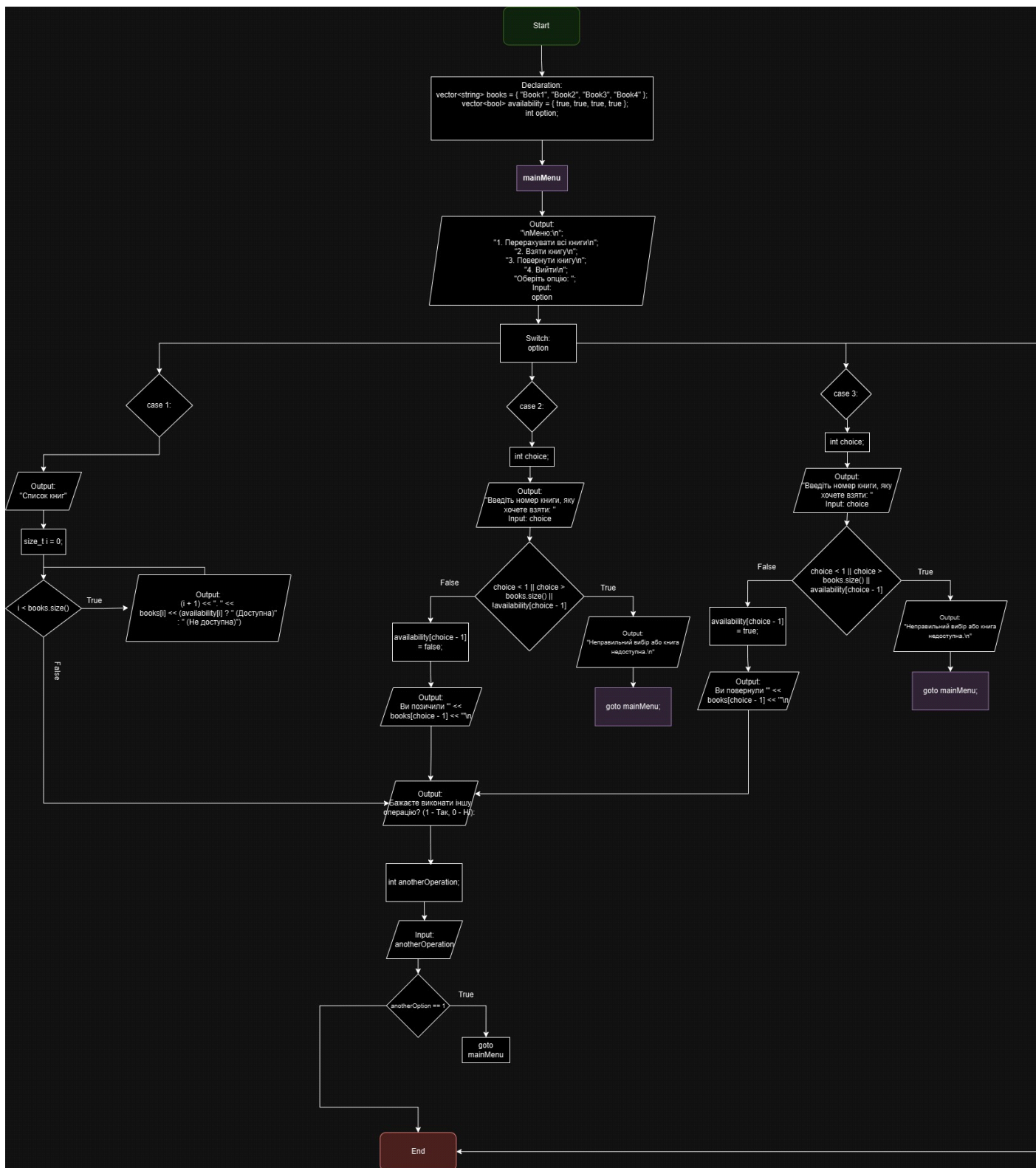
1. Перерахувати всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти

Оберіть опцію: 2

Введіть номер книги, яку хочете взяти: 3

Ви позичили 'Book3'

Бажаєте виконати іншу операцію? (1 - Так, 0 - Ні): 0



Завдання № 7

Requirements :

Self Practise Task (Algotester)

Time:

Expected: 1h

Spent: ~2h

```

#include <iostream>
#include <vector>
#include <numeric>
#include <limits> // для перевірки на переповнення

using namespace std;

int gcd(int a, int b) {
    int maxV = max(a,b), minV = min(a,b);
    a = maxV, b = minV;
    while (b != 0) {
        a %= b;
        swap(a, b);
    }
    return a;
}

int gcd_multiple(const vector<int>& numbers) {
    int result = numbers[0];
    for (int i = 1; i < numbers.size(); i++) {
        result = gcd(result, numbers[i]);
    }
    return result;
}

int main() {
    int n;

    if (!(cin >> n) || n < 1 || n > 1000) {
        cout << -1;
        return 0;
    }

    vector<int> votes(n);

    for (int i = 0; i < n; i++) {
        if (!(cin >> votes[i]) || votes[i] < 1 || votes[i] > 1000000000) {
            cout << "-1";
            return 0;
        }
    }

    int gcd_val = gcd_multiple(votes);

    long long total_seats = 0;
    for (int i = 0; i < n; i++) {
        total_seats += static_cast<long long>(votes[i]) / gcd_val;
    }

    cout << total_seats << endl;

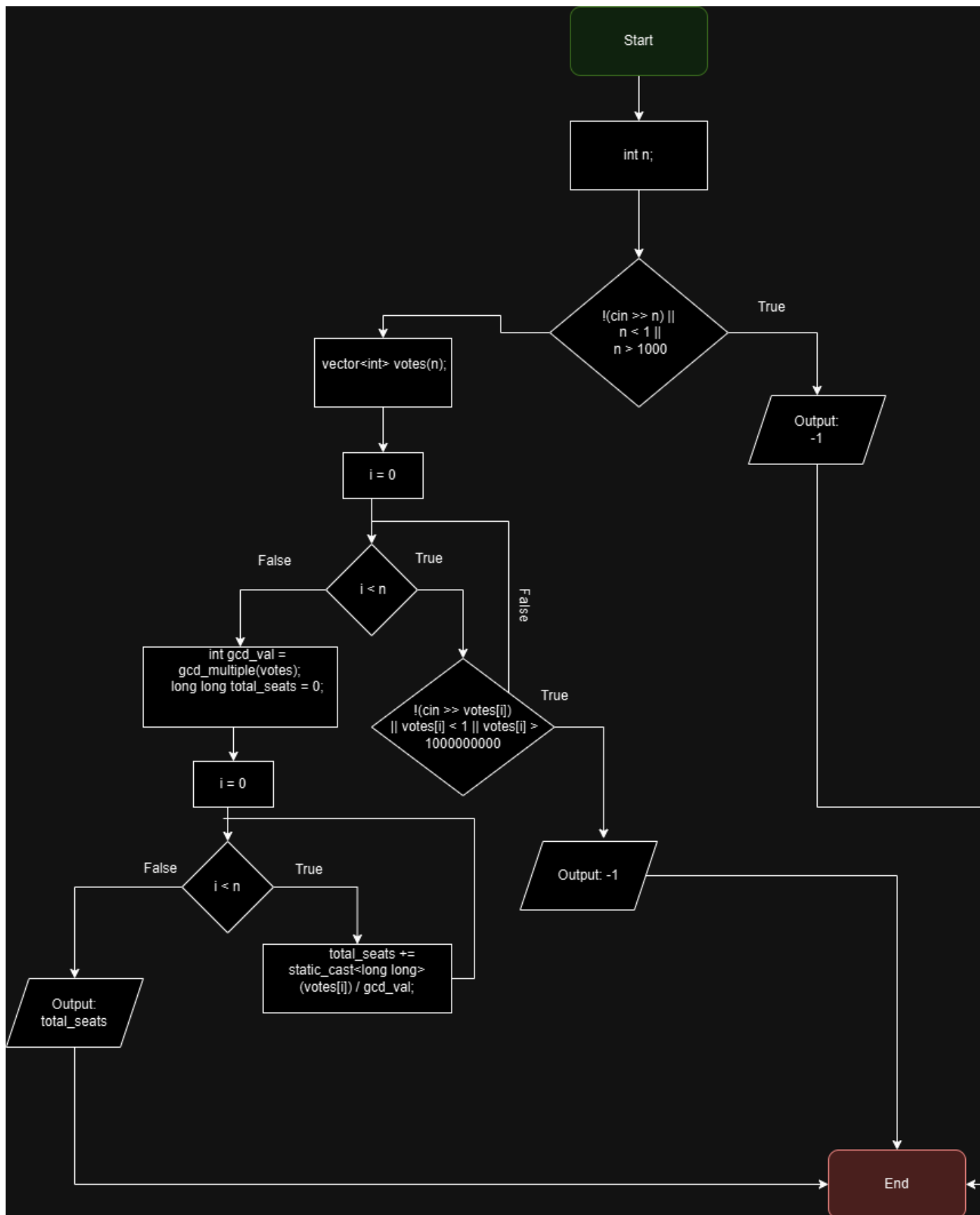
    return 0;
}

```


3

78 94 34

103



Trello & Configuration :



To Do



Nazar Kryvycho - Epic 3 Task 9 -
Results Evaluation and Release



+ Add a card



In Process



Nazar Kryvycho- Epic 3 Task 8 -
Result Documentation Report and
Outcomes Placement Activities (Docs
and Programs on GitHub)



+ Add a card



Done



Nazar Kryvycho - Epic 3 Task 3 -
Lab# programming: VNS Lab 2



Nazar Kryvycho - Epic 3 Task 7 -
Practice# programming: Self
Practice Task



Nazar Kryvycho - Epic 3 Task 6 -
Practice# programming: Class
Practice Task



Nazar Kryvycho - Epic 3 Task 4 -
Lab# programming: VNS Lab 3



Nazar Kryvycho - Epic 3 Task 5 -
Lab# programming: VNS Lab 7



Nazar Kryvycho - Epic 3 Task 1 -
Theory Education Activities



Nazar Kryvycho - Epic 3 Task 2 -
Requirements management
(understand tasks) and design
activities (draw flow diagrams and
estimate tasks 3-7)



John Black - Epic 3 Task 1 - Theory
Education Activities



John Black - Epic 3 Task 2 -
Requirements management
(understand tasks) and design
activities (draw flow diagrams and
estimate tasks 3-7)



John Black - Epic 3 Task 3 - Lab#
programming: VNS Lab 2



Pull Request: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/121

Висновок:

В результаті виконання цієї роботи я навчився ефективно використовувати цикли та вкладені цикли для реалізації повторюваних операцій у програмуванні. Опанував принципи роботи функцій, їх перевантаження, а також простір імен для покращення структури та організації коду. Навчився використовувати функції з змінною кількістю параметрів, що дозволило створювати більш гнучкі програми. Також засвоїв основи рекурсії для вирішення складних задач та навчився застосовувати вбудовані функції для полегшення стандартних операцій.