

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

Виконала:

Студентка групи ШІ-11
Андрусишин Соломія Володимирівна

Тема роботи: Ознайомлення з циклами та їх видами: for, while, do-while, та їх значенням у програмуванні. Вивчення способів управління виконанням циклів за допомогою операторів break і continue, а також передчасного завершення циклів. Аналіз вкладених циклів та їх застосування для вирішення складних завдань. Ознайомлення з основами функцій у C++: визначення, параметри, повернення значень, перевантаження функцій та робота з просторами імен. Вивчення розширених можливостей функцій, таких як еліпсис для змінної кількості параметрів, рекурсія, та використання вбудованих функцій для оптимізації коду.

Мета роботи: : Ознайомитися з основними типами циклів у C++ (for, while, do-while), їхньою роллю та синтаксисом, а також навчитися застосовувати їх у різних завданнях. Навчитися використовувати оператори break та continue, розуміти умови завершення та передчасне завершення циклів. Засвоїти принципи створення функцій, включаючи передавання параметрів, значення за замовчуванням та повернення результатів. Розібратися з поняттям перевантаження функцій та використанням просторів імен для організації коду. Опанувати роботу з функціями зі змінною кількістю параметрів, а також принципами рекурсії та їхнього застосування у складних задачах. Ознайомитися з вбудованими функціями у C++ та їх роллю у спрощенні коду й виконанні стандартних операцій. Навчитися використовувати вкладені цикли та практично застосовувати їх для вирішення комплексних завдань у програмуванні.

Теоретичні відомості:

1. Введення в Цикли та їх Види в C++:

- Значення та роль циклів у програмуванні.
- Огляд видів циклів: for, while, do-while.
- Синтаксис та основи використання кожного типу циклу.
- Приклади базових циклів для різних задач.

2. Управління Виконанням Циклів:

- Застосування операторів break та continue.

- Умови завершення циклів.
- Передчасне завершення виконання циклу.
- Приклади та вправи з управлінням циклами.

3.Вкладені Цикли:

- Поняття та важливість вкладених циклів.
- Реалізація вкладених циклів: приклади для різних сценаріїв.
- Практичні завдання на вкладені цикли.

4.Основи Функцій у C++:

- Визначення та оголошення функцій.
- Параметри функцій: передача за значенням і за посиланням.
- Параметри за замовчуванням.
- Повернення значень з функцій.
- Приклади створення та використання функцій.

5.Перевантаження Функцій та Простір Імен:

- Концепція перевантаження функцій.
- Правила та приклади перевантаження функцій.
- Поняття та використання просторів імен.
- Вкладені простори імен (C++ 17)
- Роль просторів імен у організації коду.

6.Розширені Можливості Функцій:

- Функції зі змінною кількістю параметрів (еліпсис): синтаксис та приклади.
- Область видимості функції – static, extern.
- Рекурсія: основи, приклади рекурсивних функцій та їх аналіз.
- Передача масивів та об'єктів як параметрів.
- Повернення масивів та об'єктів з функцій.

7.Вбудовані Функції в C++:

- Огляд вбудованих функцій у C++.
- Приклади використання стандартних функцій у програмуванні.
- Роль вбудованих функцій у спрощенні коду.
- Практичні завдання для розуміння вбудованих функцій.

Індивідуальний план опрацювання теорії:

1. Введення в Цикли та їх Види в C++:

- Джерела:

https://youtu.be/zBtcqNdiRf4?si=KSx_SA93IH9-tOpm

<https://youtu.be/ckJtOMcIxyU?si=0issnIWvpCX70uLh>

https://youtu.be/pBhaBdXWMmU?si=S_XFjyHXOWJGaLxF

<https://youtu.be/QXaSSlJVor8?si=uGp7gpgthHCnuSlw>

https://youtu.be/7ZEAxFb_J8s?si=TkiqEtg5oPu3xNYA

- Висновок: Отже, з цих теоретичних відомостей я дізналась, для чого потрібний цикл (цикл дозволяє використовувати певний блок коду, якусь кількість разів, поки задається ця умова.)

Також дізналась основні типи циклів і їхню різницю між собою:

- *for* (цикл із лічильником): Використовується, коли кількість ітерацій відома заздалегідь.
- *while* (цикл із передумовою): Виконується, поки умова істинна.
- *do-while* (цикл із постумовою): Виконує тіло циклу принаймні один раз, оскільки перевірка умови відбувається після виконання.

2. Управління Виконанням Циклів:

- Джерела:

<https://youtu.be/rj1OLsBKazA?si=f2zY-9TGLfLqC0ae>

https://youtu.be/UY295pIdeoQ?si=9Xsw7PdbLfoU_N95

- Висновок: Отже, з цих теоретичних відомостей я дізналась, для чого застосовують *break* та *continue* і де їх застосовують
 - *Break* - зупиняє виконання поточного циклу (або блоку *switch*) і передає керування на наступний рядок після циклу чи блоку.
 - *Continue* - пропускає залишок тіла циклу для поточної ітерації та переходить до наступної.

3. Основи Функцій у C++:

- Джерела:
<https://youtu.be/G8P6SvdqU9s?si=fT7y9hyJB4YOWjRK>
- Висновок: Отже, з цих теоретичних відомостей я дізналась, для чого застосовують потрібні функції, і яка з них користь.
 - Функція - це блок коду, який виконує певну задачу. Вона може приймати вхідні дані (аргументи), виконувати операції та повертати результат.
 - Чим відрізняється функція *void*.
 - Функції з параметрами і без;
 - Переривання роботи функції(*return, throw*).

4. Перевантаження Функцій та Простір Імен:

- Джерела:
<https://youtu.be/3KJfisev6SI?si=ar3QfkprzSKgM7o>
<https://acode.com.ua/urok-108-perevantazhennya-funktsij/>
- Висновок: **Перевантаження функцій** — це можливість визначати декілька функцій з одним і тим же ім'ям, але з різними параметрами.
 - Типи повернення в перевантаженні функцій
 - Переваги та недоліки;

5. Функції зі змінною кількістю параметрів (еліпсис):

- Джерела:
<https://youtu.be/3KJfisev6SI?si=ar3QfkprzSKgM7o>
- Висновок: Отже, *еліпсис* у програмуванні використовується для позначення змінної кількості аргументів у функціях. Він дозволяє передавати невизначену кількість параметрів у функцію.
 - В *еліпсисі* немає ніяких імен змінних. Замість цього ми отримуємо доступ до значень через спеціальний тип — *va_list*. Про *va_list* можна думати, як про вказівник, який вказує на масив з еліпсисом. Спочатку ми оголошуємо змінну *va_list*, яку називаємо просто *list* для зручності використання.

- Використовуємо ***va_arg*** для отримання параметрів з еліпсиса. Перший параметр - це ***va_list***, який ми використовуємо. Другий параметр - це очікуваний тип параметрів.
- Виконуємо очищення ***va_list***, коли вже зробили все необхідне за допомогою ***va_end***.

6. Розширені Можливості Функцій:

- Джерела:
https://youtu.be/V7q9w_s0nns?si=NaX_hmrTGYxq_Khr
- Висновок: ***Рекурсія*** — це техніка програмування, коли функція викликає сама себе безпосередньо або опосередковано, щоб вирішити певну задачу.

7. Вбудовані Функції в C++:

- Джерела:
<https://acode.com.ua/urok-107-vbudovani-funktsiyi/>
- Висновок: ***Вбудовані функції*** - це стандартні функції, які надаються бібліотеками і доступні для використання без необхідності додаткової реалізації. Вони виконують базові й часто використовувані операції, такі як робота з математичними виразами, обробка рядків, введення/виведення.

Виконання роботи:

Завдання №1

(VNS Lab2 Task1) - Використовуючи оператор циклу, знайти суму елементів, зазначених у конкретному варіанті. Результат надрукувати, надавши відповідний заголовок.

6) Знайти суму ряду з точністю $\epsilon=0.0001$, загальний член якого

$$a_n = \frac{1}{2^n} + \frac{1}{3^n}$$

Завдання №2

(VNS Lab3 Task1) - Для x , що змінюється від a до b з кроком $(b-a)/k$, де $(k=10)$, обчислити функцію $f(x)$, використовуючи її розклад в степеневий ряд у двох випадках:

а) для заданого n ;

б) для заданої точності ε ($\varepsilon=0.0001$). Для порівняння знайти точне значення функції.

$y = e^{x \cos \frac{\pi}{4}} \cdot \cos(x \sin \frac{\pi}{4})$	$0,1 \leq x \leq 1$	25	$S = 1 + \frac{\cos \frac{\pi}{4}}{1!} x + \dots + \frac{\cos n \frac{\pi}{4}}{n!} x^n$
---	---------------------	----	---

Завдання №3

(VNS Lab7 Task1) –

Розв’язати зазначене у варіанті завдання, використовуючи функції зі змінною кількістю параметрів.

-Написати функцію `min` зі змінною кількістю параметрів, що знаходить мінімальне із чисел типу `int`. Написати викликаючу функцію `main`, що звертається до функції `min` не менше трьох разів з кількістю параметрів 5, 10, 12.

Завдання №4

(VNS Lab7 Task2) –

Написати перевантажені функції й основну програму, що їх викликає.

а) для віднімання дійсних чисел;

б) для віднімання комплексних чисел.

Завдання №5

(Class Practice Work)

Ви створюєте просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

Програма повинна вміти:

- Перерахувати всі книги.
- Дозволити взяти книгу (за наявності).
- Дозволити повернення книги.

Структури даних

- Використовуйте масив або вектор для зберігання назв книг.

- Використовуйте інший масив або вектор для збереження стану доступності кожної книги.

Мета Задачі:

Навчитися користуватися операторами циклів та функцією переходу на мітку:

1. `for() { ... }`
2. `for each`
3. `while() { ... }`
4. `do { ... } while()`
5. `go to`

Вимоги:

1. `while`: продовжувати працювати, доки користувач не вирішить вийти.
2. `do while`: Після кожної операції (позичити, повернути, перерахувати) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.
3. `for`: список усіх книг за допомогою циклу.
4. `for each`: перевірити наявність кожної книги.
5. `goto`: якщо користувач вводить неправильний вибір, використовуйте `goto`, щоб перенаправити його до головного меню.

Завдання №6 (Self Practice Work)

День програміста

Обмеження: 2 сек., 256 MiB

Нарешті ми його дочекалися, 256-го дня в році (дня програміста)... Кожен зі студентів факультету прикладної математики та інформатики святкували його по-різному: дехто вдома, дехто в «Picasso», дехто в гуртожитку... Не дивно, що Зеник із Марічкою залишилися святкувати його в гуртожитку. Вони організували «mega party» :-). Свято вдалося...

Наступного дня, гуляючи з Марічкою, Зенику стало цікаво, скільки ж було випито різної випивки?!

Марічка змогла згадати nn назв випивок, а Зеник — mm .

Вам потрібно написати програму, яка порахує скільки ж було різної випивки на святі, яку змогли згадати Зеник з Марічкою.

Вхідні дані

У першому рядку дано цілі числа nn і mm .

У наступних nn рядках дано назви випивок, які змогла згадати Марічка.

В наступних mm рядках дано назви випивок, які зміг згадати Зеник.

Вихідні дані

Єдине ціле число — кількість різної випивки на святі.

Обмеження

$0 \leq n, m \leq 777$

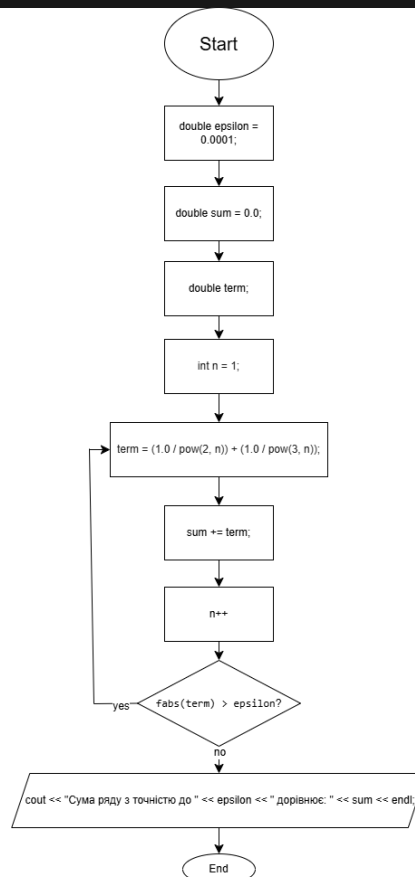
$1 \leq |name_i| \leq 7$ (лише маленькі латинські літери),

$|a|$ — довжина aa .

Завдання №1 (VNS Lab2 Task1)

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main()
6  {
7      const double epsilon = 0.0001;
8      double sum = 0.0;
9      double term;
10     int n = 1;
11
12     do
13     {
14         term = (1.0 / pow(2, n)) + (1.0 / pow(3, n));
15         sum += term;
16         n++;
17     }
18     while (fabs(term) > epsilon);
19
20     cout << "Сума ряду з точністю до " << epsilon << " дорівнює: " << sum << endl;
21 }
22
```

Сума ряду з точністю до 0.0001 дорівнює: 1.49994
PS C:\Users\Solomia>



Завдання №2 (VNS Lab3 Task1)

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  const double PI = 3.141592653589793;
6  const double TOLERANCE = 0.0001;
7
8  double analyticalFunction(double x)
9  {
10     double angle_cos = cos(PI / 4);
11     double angle_sin = sin(PI / 4);
12     return exp(x * angle_cos) * cos(x * angle_sin);
13 }
14
15 double computeCoefficient(double x, int n) {
16     double coefficient = 1.0;
17     for (int i = 1; i <= n; i++)
18     {
19         coefficient *= (x / i); // C_n = x^n / n!
20     }
21     return coefficient;
22 }
23
24 double computeSeriesTerm(double x, int n) {
25     double angle_term = cos(n * PI / 4); // cos(n * π / 4)
26     double coefficient = computeCoefficient(x, n); // x^n / n!
27     return angle_term * coefficient; // S_n = cos(n * π / 4) * C_n
28 }
29
30 int main()
31 {
32     double start = 0.1, end = 1.0;
33     int steps = 10;
34     double increment = (end - start) / steps;
35
36     for (double x = start; x <= end; x += increment) {
37         double partialSumFixed = 1.0;
38         double partialSumTolerance = 1.0;
39         double currentTerm;
40
41         for (int i = 1; i <= 25; i++) {
42             currentTerm = computeSeriesTerm(x, i);
43             partialSumFixed += currentTerm;
44         }
45     }
```

```

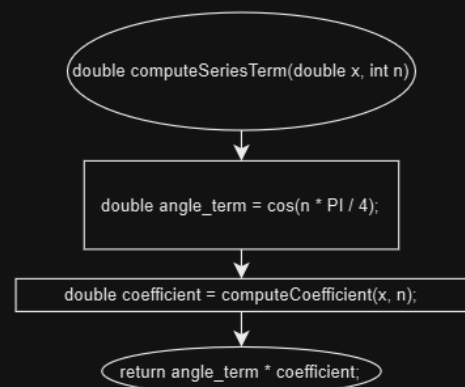
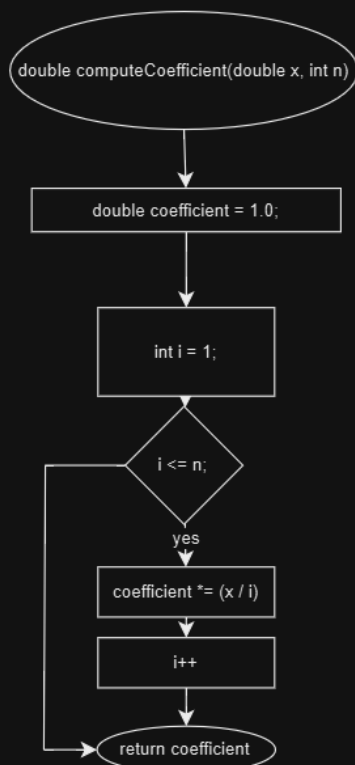
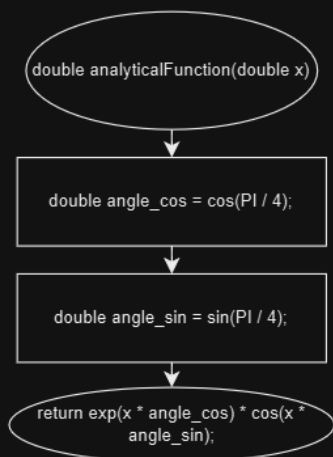
46     int termIndex = 1;
47     while (true) {
48         currentTerm = computeSeriesTerm(x, termIndex);
49         if (abs(currentTerm) < TOLERANCE) break;
50         partialSumTolerance += currentTerm;
51         termIndex++;
52     }
53
54     cout << "X = " << x
55          << "\tSN = " << partialSumFixed
56          << "\tSE = " << partialSumTolerance
57          << "\tY = " << analyticalFunction(x)
58          << endl;
59 }
60
61 return 0;
62 }
63

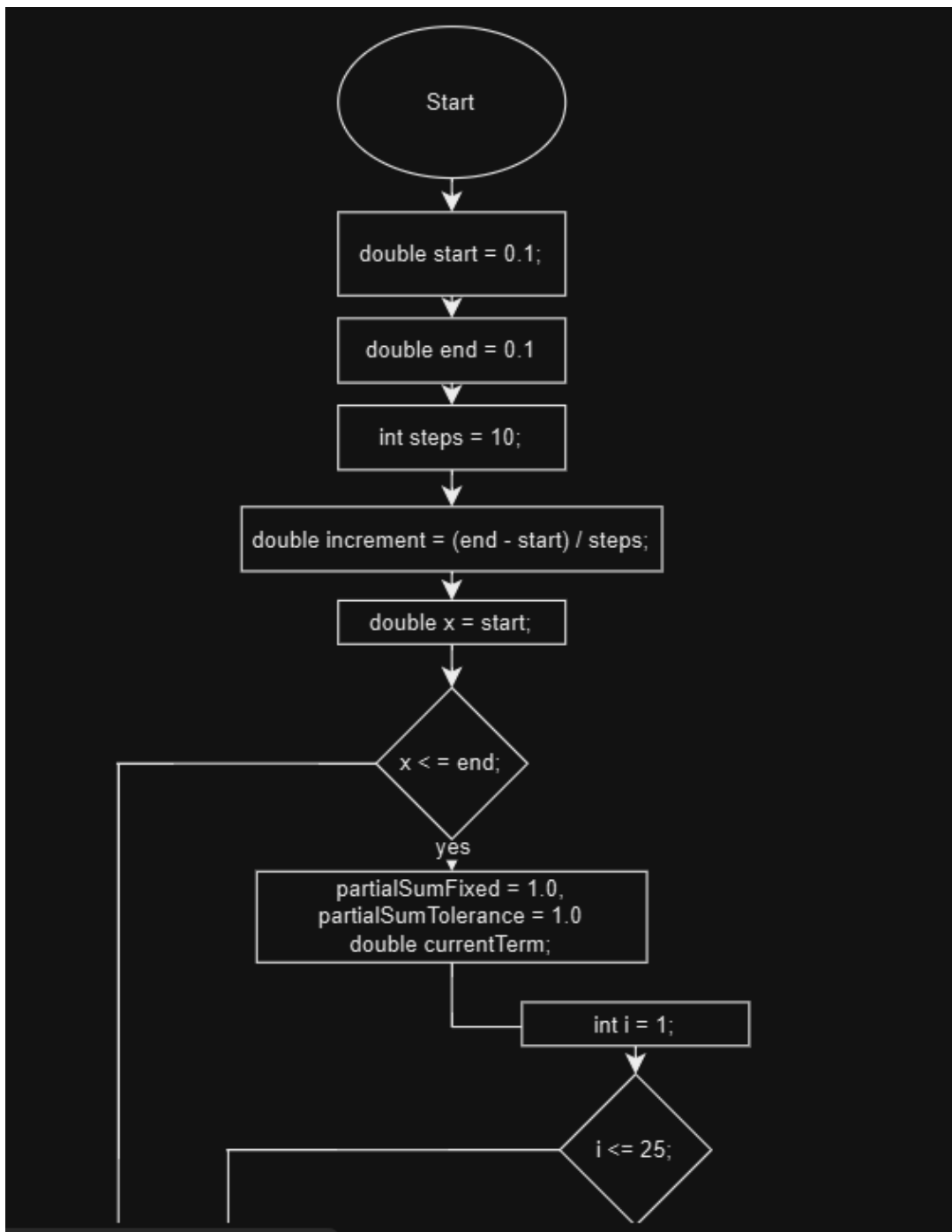
```

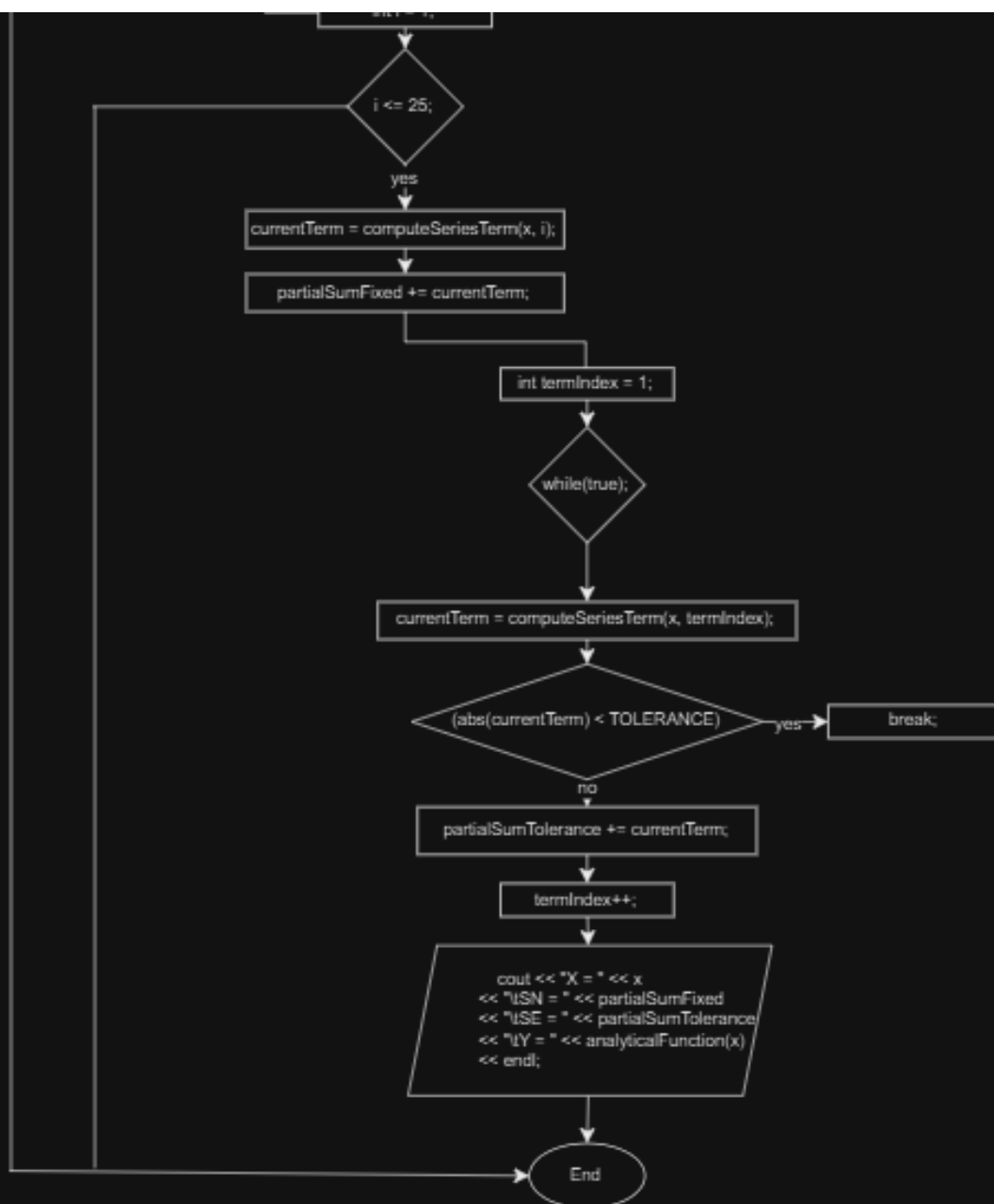
```

X = 0.1 SN = 1.07059 SE = 1.07071 Y = 1.07059
X = 0.19 SN = 1.13349 SE = 1.13435 Y = 1.13349
X = 0.28 SN = 1.19514 SE = 1.19799 Y = 1.19514
X = 0.37 SN = 1.25484 SE = 1.26163 Y = 1.25484
X = 0.46 SN = 1.31181 SE = 1.32527 Y = 1.31181
X = 0.55 SN = 1.36519 SE = 1.38891 Y = 1.36519
X = 0.64 SN = 1.41404 SE = 1.45255 Y = 1.41404
X = 0.73 SN = 1.45731 SE = 1.51619 Y = 1.45731
X = 0.82 SN = 1.49387 SE = 1.57983 Y = 1.49387
X = 0.91 SN = 1.52249 SE = 1.64347 Y = 1.52249
X = 1 SN = 1.54186 SE = 1.70711 Y = 1.54186
PS C:\Users\Solomia>

```



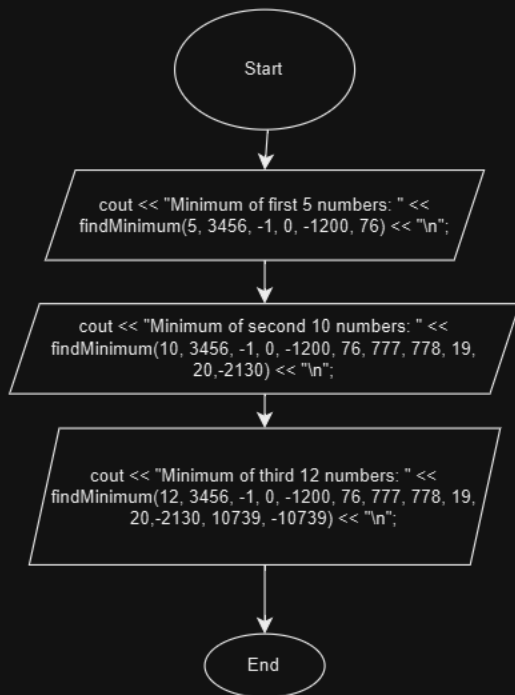


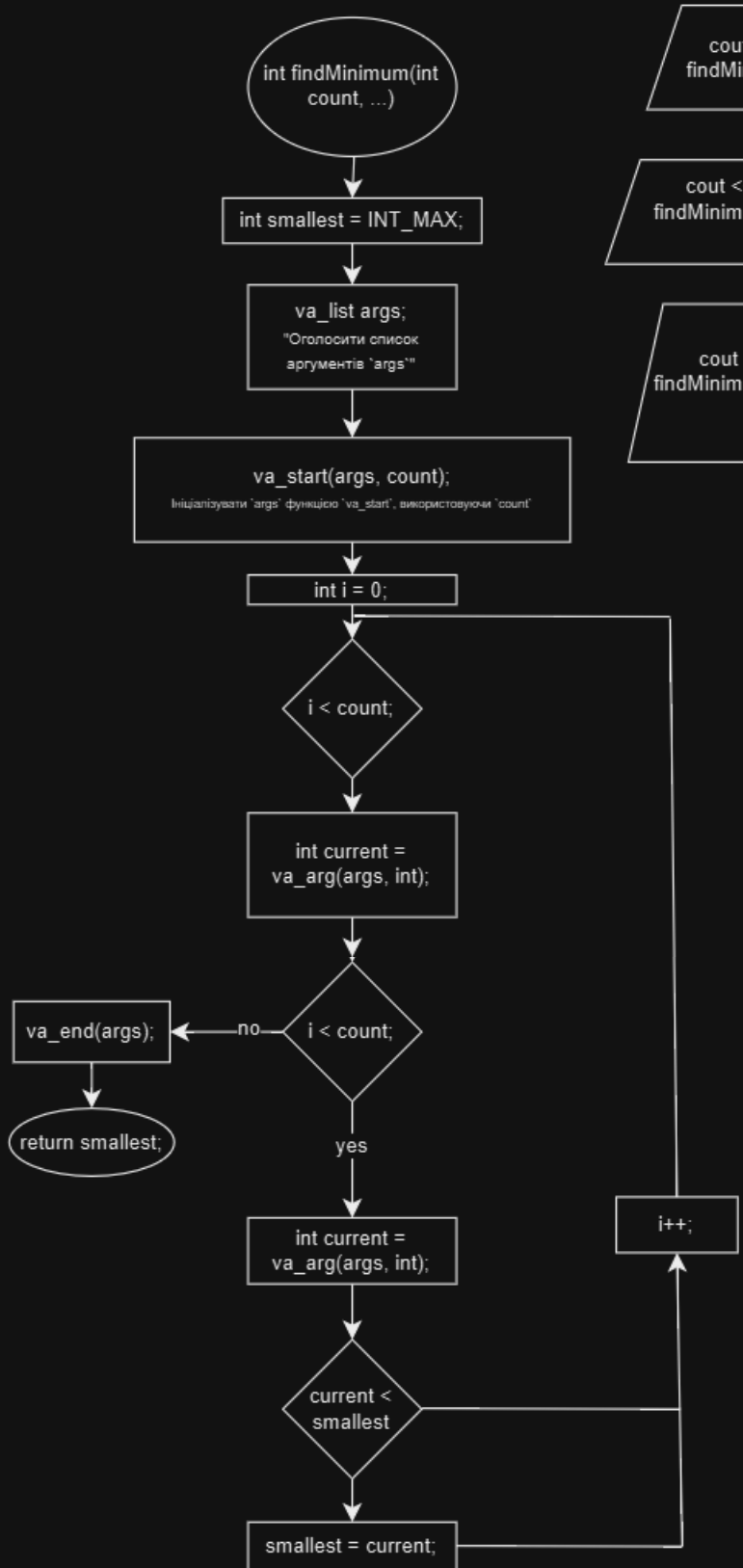


Завдання №3 (VNS Lab7 Task1)

```
1  #include <iostream>
2  #include <cstdlib>
3
4  using namespace std;
5
6  int findMinimum(int count, ...)
7  {
8      int smallest = INT_MAX;
9      va_list args;
10     va_start(args, count);
11
12     for (int i = 0; i < count; ++i) {
13         int current = va_arg(args, int);
14         if (current < smallest) {
15             smallest = current;
16         }
17     }
18
19     va_end(args);
20     return smallest;
21 }
22
23 int main() {
24     cout << "Minimum of first 5 numbers: " << findMinimum(5, 3456, -1, 0, -1200, 76) << "\n";
25     cout << "Minimum of second 10 numbers: " << findMinimum(10, 3456, -1, 0, -1200, 76, 777, 778, 19, 20, -2130) << "\n";
26     cout << "Minimum of third 12 numbers: " << findMinimum(12, 3456, -1, 0, -1200, 76, 777, 778, 19, 20, -2130, 10739, -10739) << "\n";
27
28     return 0;
29 }
30
```

```
Minimum of first 5 numbers: -1200
Minimum of second 10 numbers: -2130
Minimum of third 12 numbers: -10739
PS C:\Users\Solomia>
```





cout
findMir

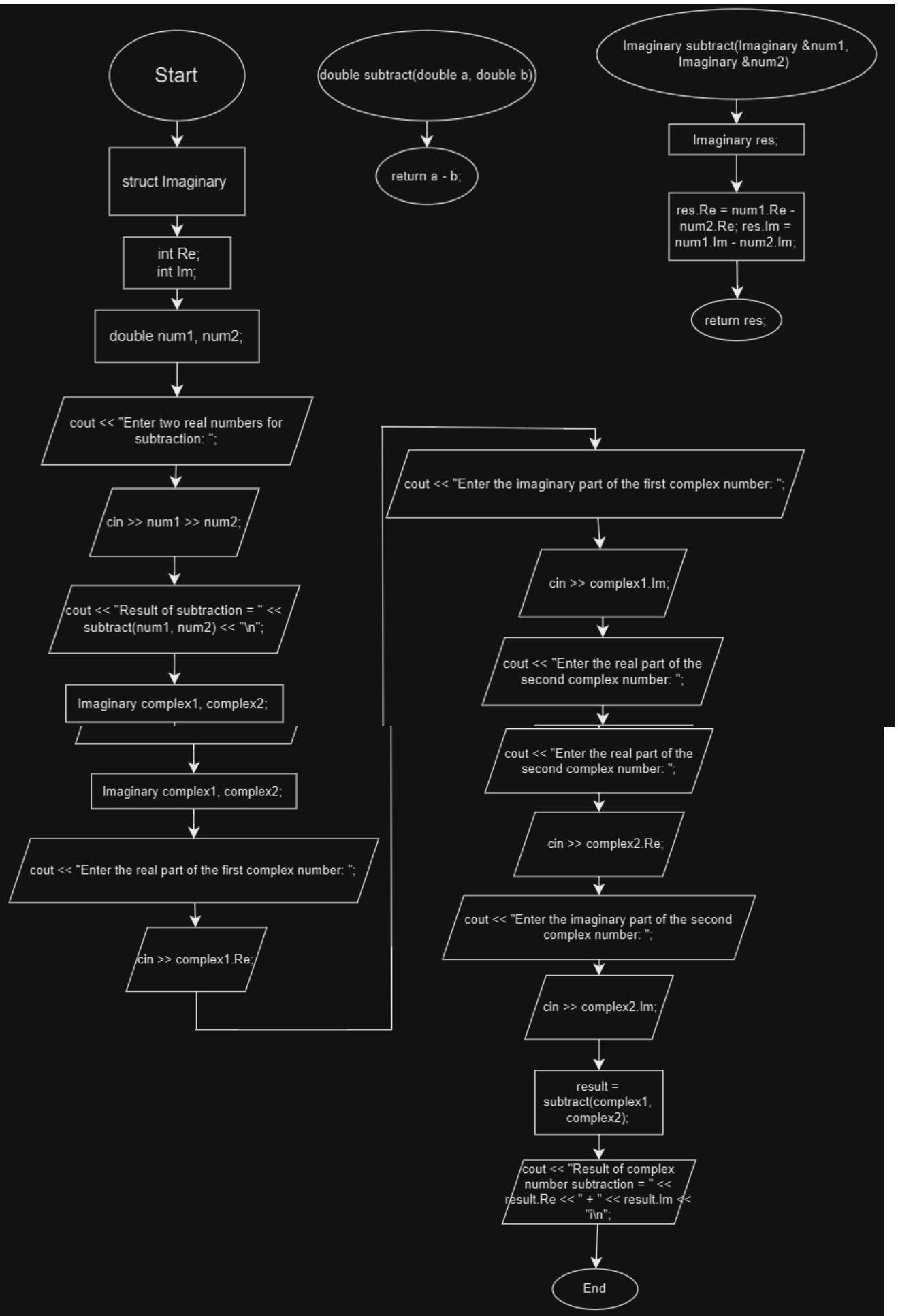
cout <<
findMinimu

cout <
findMinimu
2

Завдання №4 (VNS Lab7 Task2)

```
4  struct Imaginary
5  {
6      int Re;
7      int Im;
8  };
9
10 double subtract(double a, double b) {
11     return a - b;
12 }
13
14 Imaginary subtract(Imaginary &num1, Imaginary &num2)
15 {
16     Imaginary res;
17     res.Re = num1.Re - num2.Re;
18     res.Im = num1.Im - num2.Im;
19     return res;
20 }
21
22 int main()
23 {
24     double num1, num2;
25     cout << "Enter two real numbers for subtraction: ";
26     cin >> num1 >> num2;
27     cout << "Result of subtraction = " << subtract(num1, num2) << "\n";
28
29     Imaginary complex1, complex2;
30     cout << "Enter the real part of the first complex number: ";
31     cin >> complex1.Re;
32     cout << "Enter the imaginary part of the first complex number: ";
33     cin >> complex1.Im;
34
35     cout << "Enter the real part of the second complex number: ";
36     cin >> complex2.Re;
37     cout << "Enter the imaginary part of the second complex number: ";
38     cin >> complex2.Im;
39
40     Imaginary result = subtract(complex1, complex2);
41     cout << "Result of complex number subtraction = " << result.Re << " + " << result.Im << "i\n";
42
43     return 0;
44 }
```

```
Enter two real numbers for subtraction: 6 5
Result of subtraction = 1
Enter the real part of the first complex number: -2 5
Enter the imaginary part of the first complex number: Enter the real part of the second complex number: 9 7
Enter the imaginary part of the second complex number: Result of complex number subtraction = -11 + -2i
PS C:\Users\Solomia>
```



Завдання №5 (Class Practice Work)

```
1  #include <iostream>
2
3  using namespace std;
4
5  const int MAX_BOOKS = 4;
6
7
8  void listBooks(const int available[])
9  {
10     cout << "\nList of books in the library:\n";
11     for (int i = 0; i < MAX_BOOKS; i++) {
12         cout << i + 1 << ". Book " << i + 1 << " - ";
13
14         if (available[i] == 1)
15         {
16             cout << "Available";
17         }
18         else
19         {
20             cout << "Not available";
21         }
22         cout << endl;
23     }
24 }
25
26
27 void borrowBook(int available[])
28 {
29     int choice;
30     cout << "Enter number of book , that you want take: ";
31     cin >> choice;
32
33
34     if (choice > 0 && choice <= MAX_BOOKS)
35     {
36         if (available[choice - 1] == 1) {
37             available[choice - 1] = 0;
38             cout << "You take this book:\n";
39         }
40         else
41         {
42             cout << "This book is already taken :\n";
43         }
44     }
45     else
```

```

46     {
47         cout << "Unknowing number of book.\n";
48     }
49 }
50
51
52 void returnBook(int available[])
53 {
54     int choice;
55     cout << "Enter number of book, that you want return: ";
56     cin >> choice;
57
58
59     if (choice > 0 && choice <= MAX_BOOKS) {
60         if (available[choice - 1] == 0)
61         {
62             available[choice - 1] = 1;
63             cout << "You return this bool.\n";
64         }
65         else
66         {
67             cout << "This book is in library.\n";
68         }
69     }
70     else
71     {
72         cout << "Unknowing number of book.\n";
73     }
74 }
75
76 int main() {
77
78     int available[MAX_BOOKS] = { 1, 1, 1, 1 };
79
80     int choice;
81
82     do {
83         main_menu:
84         cout << "\nMenu:\n";
85         cout << "1. Look all books\n";
86         cout << "2. Take a book\n";
87         cout << "3. Return a book\n";
88         cout << "4. Exit\n";

```

```
88     cout << "4. Exit\n";
89     cout << "Choose operation: ";
90     cin >> choice;
91
92
93     if (choice == 1)
94     {
95         listBooks(available);
96     }
97     else if (choice == 2)
98     {
99         borrowBook(available);
100    }
101    else if (choice == 3)
102    {
103        returnBook(available);
104    }
105    else if (choice == 4)
106    {
107        cout << "Goodbye!\n";
108    }
109    else
110    {
111
112        cout << "Wrong choice. Try again.\n";
113        goto main_menu;
114    }
115
116
117    if (choice != 4) {
118        char continueChoice;
119        cout << "\nDo another operation? (y/n): ";
120        cin >> continueChoice;
121
122
123        if (continueChoice != 'y' && continueChoice != 'Y') break;
124    }
125    } while (choice != 4);
126
127    return 0;
128 }
129
```

Menu:

1. Look all books
2. Take a book
3. Return a book
4. Exit

Choose operation: 1

List of books in the library::

1. Book 1 - Available
2. Book 2 - Available
3. Book 3 - Available
4. Book 4 - Available

Do another operation? (y/n): y

Menu:

1. Look all books
2. Take a book
3. Return a book
4. Exit

Choose operation: 2

Enter number of book , that you want take: 1

You take this book:

Do another operation? (y/n): y

Menu:

1. Look all books
2. Take a book
3. Return a book
4. Exit

Choose operation: 2

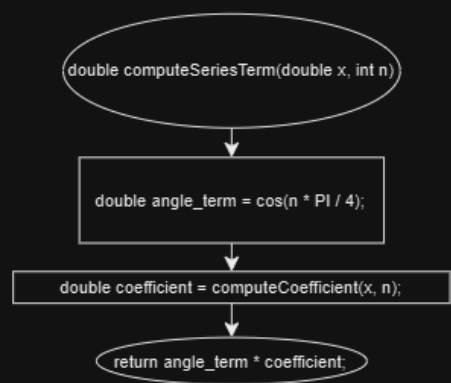
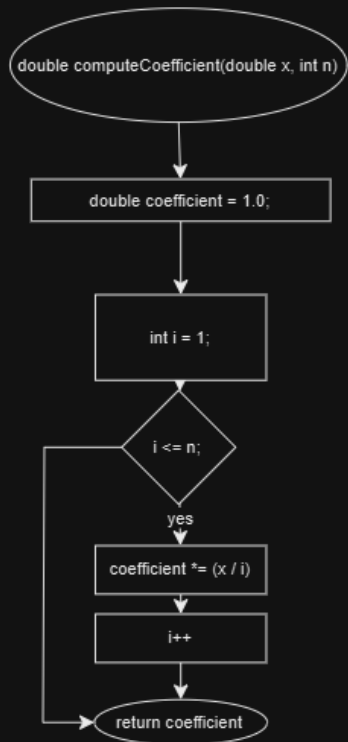
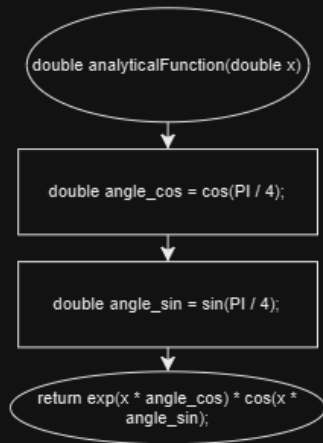
Enter number of book , that you want take: 1

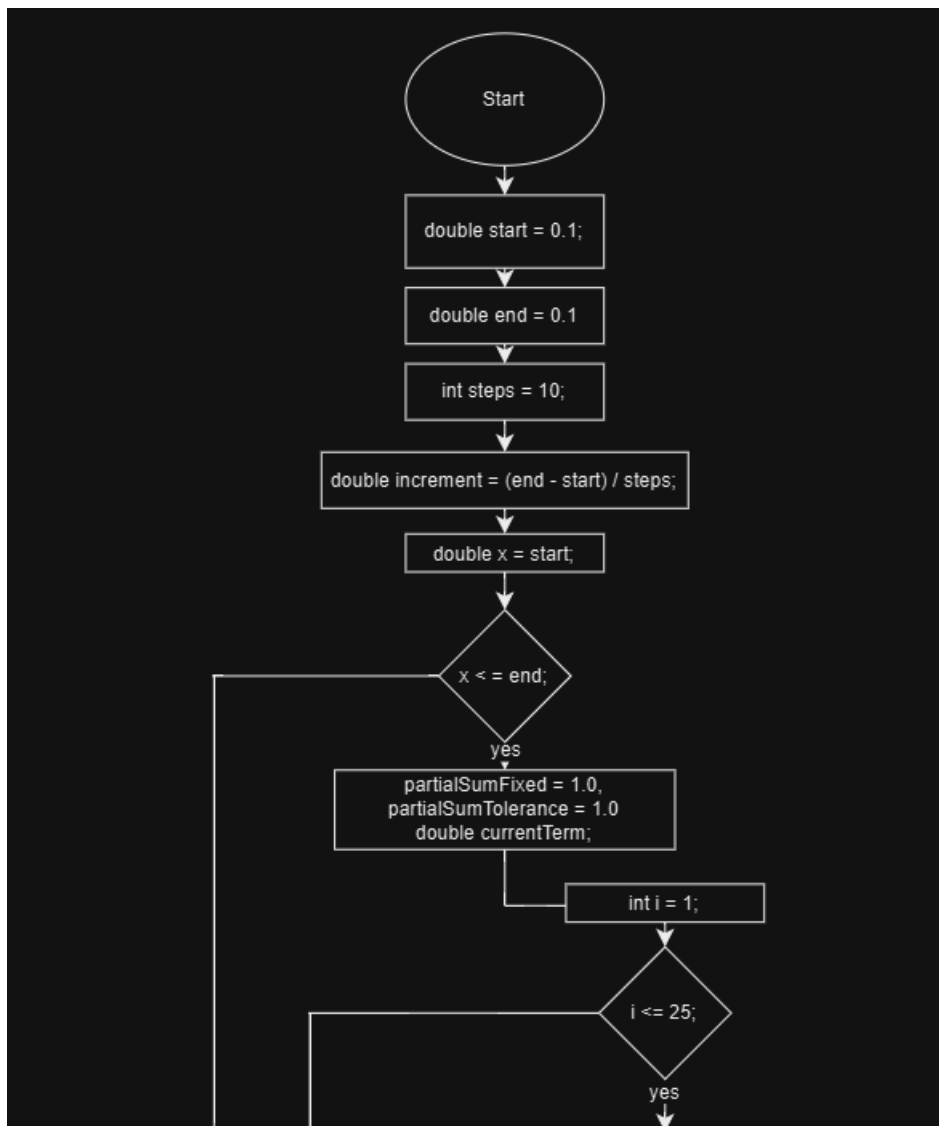
This book is already taken :

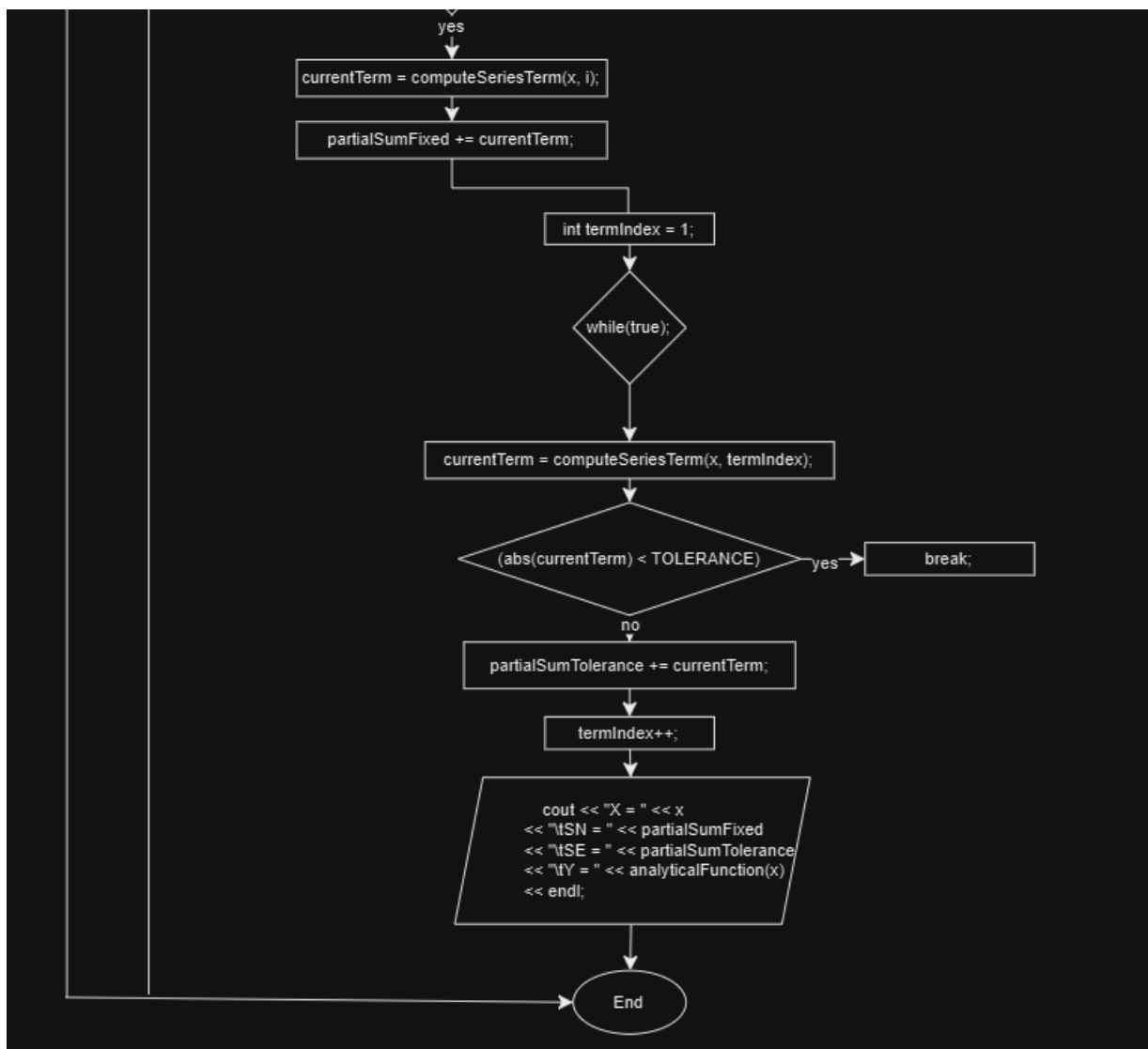
Do another operation? (y/n): y

Menu:

1. Look all books
2. Take a book
3. Return a book







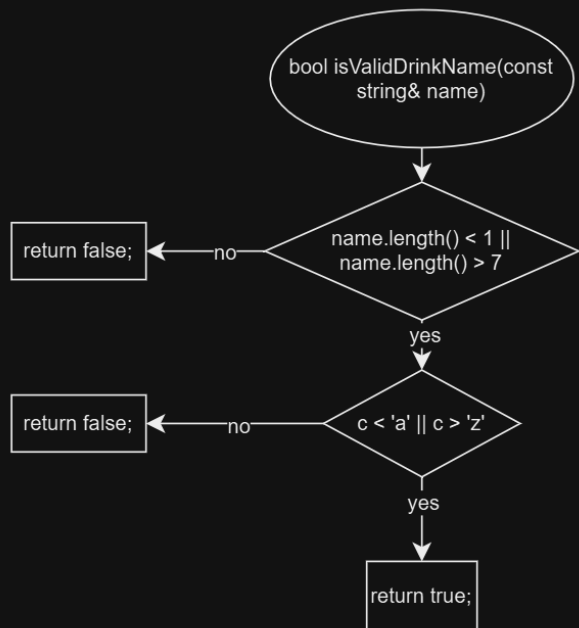
Завдання №6
(Self Practice Work)

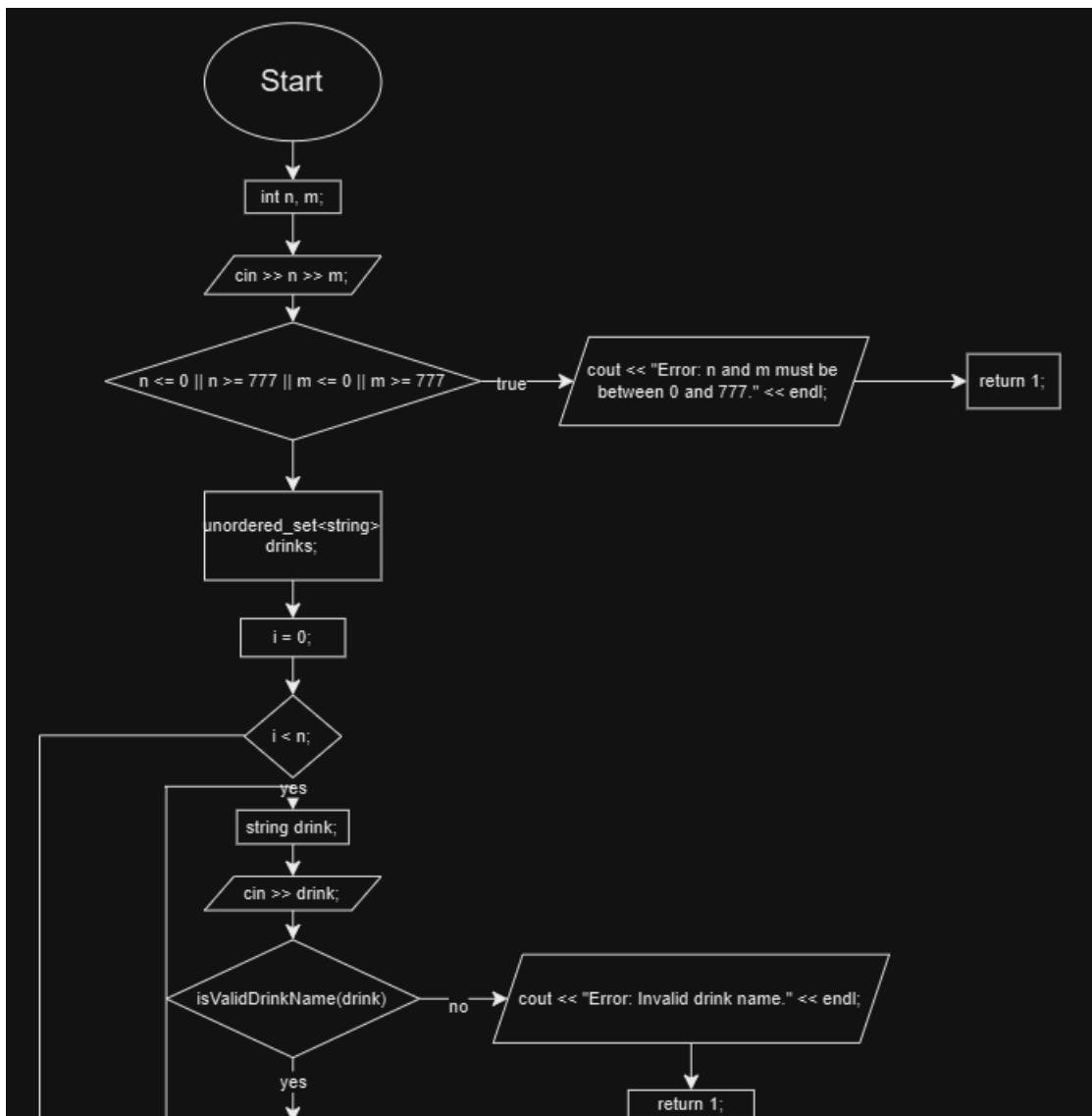
```

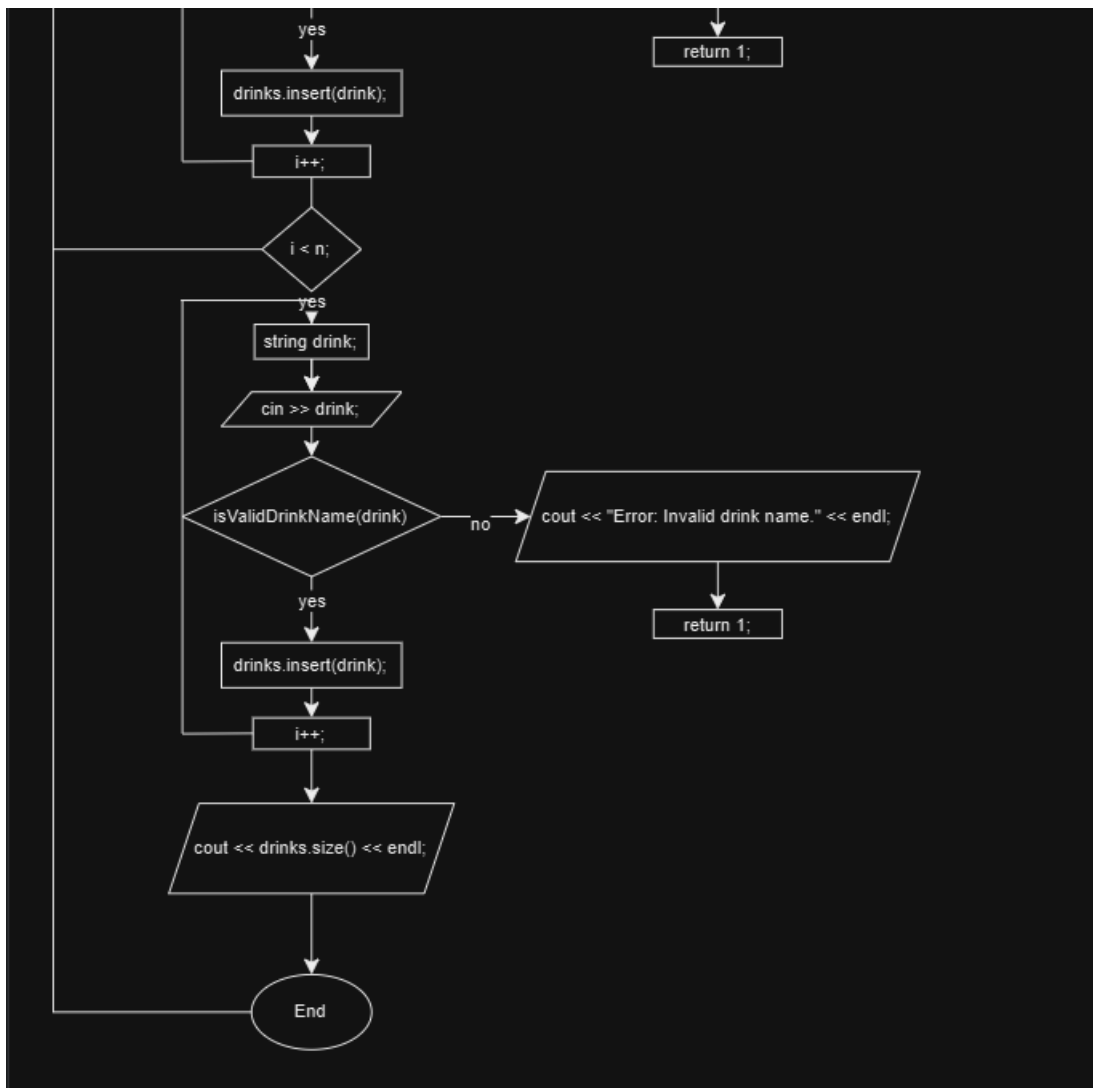
1  #include <iostream>
2  #include <unordered_set>
3  #include <string>
4
5  using namespace std;
6
7  bool isValidDrinkName(const string& name) {
8      if (name.length() < 1 || name.length() > 7)
9          {
10             return false;
11         }
12     for (char c : name) {
13         if (c < 'a' || c > 'z')
14             {
15                 return false;
16             }
17     }
18
19     return true;
20 }
21
22 int main() {
23     int n, m;
24     cin >> n >> m;
25
26     if (n <= 0 || n >= 777 || m <= 0 || m >= 777) {
27         cout << "Error: n and m must be between 0 and 777." << endl;
28         return 1;
29     }
30
31     unordered_set<string> drinks;
32
33     for (int i = 0; i < n; ++i) {
34         string drink;
35         cin >> drink;
36         if (isValidDrinkName(drink))
37             {
38                 drinks.insert(drink);
39             } else {
40                 cout << "Error: Invalid drink name." << endl;
41                 return 1;
42             }
43     }
44
45     for (int i = 0; i < m; ++i) {
46         string drink;
47         cin >> drink;
48         if (isValidDrinkName(drink)) {
49             drinks.insert(drink);
50         } else {
51             cout << "Error: Invalid drink name." << endl;
52             return 1;
53         }
54     }
55
56     cout << drinks.size() << endl;
57
58     return 0;
59 }

```

```
2 3
cola pepsi
cola pepsi water
3
PS C:\Users\Solomia> |
```







Дошка Trello:

The screenshot shows a Trello board interface with the following details:

- Board Name:** Дошка
- Columns:**
 - must do:** Contains 5 task cards. The last card has a '+ Додати картку' button.
 - in procces:** Contains 2 task cards. The last card has a '+ Додати картку' button.
 - done:** Contains 4 task cards. The last card has a '+ Додати картку' button.
- Task Cards:** All cards follow a similar naming convention, e.g., 'vns_lab_3_task_1_variant_6_solomiia_andrusyshyn'.
- Interface Elements:** Top bar includes user 'epic3', a search icon, and a 'Фільтри' (Filters) button. The bottom of the board features a desert landscape background image.

Висновок: На лабораторній: роботі №3 (еріс 3) , я ознайомилась з поняттями ,що таке функції ,цикли, як і де їх використовувати.Опрацювала як працює область видимості та як задавати глобально змінні.