

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 ІІІ-11

Гуменюк Анастасія Олександрівна

**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. Теоретичні відомості з переліком важливих тем:
  - Тема №1: Введення в цикли та їх види в C++. Вкладені цикли.
  - Тема № 2: Управління виконанням циклів.
  - Тема № 3: Основи Функцій у C++.
  - Тема № 4: Перевантаження Функцій та Простір Імен.
  - Тема № 5: Функції зі змінною кількістю параметрів (еліпсис).
  - Тема № 6: Рекурсія: основи, приклади рекурсивних функцій та їх аналіз.
  - Тема № 7: Вбудовані Функції в C++.

### **2. Індивідуальний план опрацювання теорії:**

Тема №1: Введення в цикли та їх види в C++. Вкладені цикли.

- Джерела:

- <https://acode.com.ua/urok-72-tsykl-for>
- <https://acode.com.ua/urok-70-tsykl-while/>
- <https://acode.com.ua/urok-69-operator-goto/>
- Що опрацьовано:
  - Область видимості циклу.
  - Помилка неврахованої одиниці.
  - Пропуск виразів в циклі for та запис в форматі for ( ; i<10; )
  - Тип signed int для лічильників циклу.
- Статус: Ознайомлена
- Початок опрацювання теми: 02.11.2024.
- Звершення опрацювання теми: 02.11.2024 (35хв.).

## Тема №2: Управління виконанням циклів.

- Джерела:
  - <https://acode.com.ua/urok-69-operator-goto/>
- Що опрацьовано:
  - Нескінченний цикл while та return, break, exit, goto для виходу з нього.
  - Оператор goto та лейбл.
- Статус: Ознайомлена
- Початок опрацювання теми: 02.11.2024.
- Звершення опрацювання теми: 02.11.2024 (10хв.).

## Тема №3: Основи Функцій у C++.

- Джерела:
  - <https://acode.com.ua/urok-15-funktsiyi-i-operator-return/>
  - <https://www.youtube.com/watch?v=VKQ242d-Rag>
- Що опрацьовано:
  - Особливості функцій типу void та їх (виклику) виводу.
  - Переривання роботи функції.
  - Скелет(прототип) функції.
- Статус: Ознайомлена
- Початок опрацювання теми: 02.11.2024.
- Звершення опрацювання теми: 02.11.2024 (20хв.).

## Тема №4: Перевантаження Функцій та Простір Імен.

- Джерела:
  - <https://www.youtube.com/watch?v=kYxNioENAIo>
  - <https://www.youtube.com/watch?v=3KJfisev6SI>
- Що опрацьовано:

- Переваги перевантаження функцій.
- Перевантаження зміною кількості аргументів.
- Перевантаження зміною типів аргументів.
- Перевантаження cout.
- Статус: Ознайомлена
- Початок опрацювання теми: 03.11.2024.
- Звершення опрацювання теми: 03.11.2024 (30хв.).

Тема №5: Функції зі змінною кількістю параметрів (еліпсис).

- Джерела:
  - <https://acode.com.ua/urok-117-elipsys/>
- Що опрацьовано:
  - доступ до еліпсису через va\_list.
  - Ініціалізуємо va\_list, використовуючи va\_start. Перший параметр - це список, який потрібно ініціалізувати. Другий параметр - це останній параметр, який не є еліпсисом.
  - 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.
  - Параметри еліпсиса не мають оголошень типу даних. При їх використанні компілятор повністю пропускає перевірку типів даних.
- Статус: Ознайомлена
- Початок опрацювання теми: 03.11.2024.
- Звершення опрацювання теми: 03.11.2024 (30хв.).

Тема №6: Рекурсія: основи, приклади рекурсивних функцій та їх аналіз.

- Джерела:
  - [https://www.youtube.com/watch?v=V7q9w\\_s0nns&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=54&t=31s](https://www.youtube.com/watch?v=V7q9w_s0nns&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=54&t=31s)
- Що опрацьовано:
  - Рекурсія та її завдання.
  - Недолік рекурсії щодо використання пам'яті через постійне створення змінних.
- Статус: Ознайомлена
- Початок опрацювання теми: 03.11.2024.
- Звершення опрацювання теми: 03.11.2024 (25хв.).

Тема №7: Вбудовані Функції в C++.

- Джерела:
  - <https://www.youtube.com/watch?v=ZUrdKTBRXWo>
- Що опрацьовано:
  - Для чого вбудовувати функції.
  - В яких випадках функції не можуть бути вбудованими.
- Статус: Ознайомлена
- Початок опрацювання теми: 03.11.2024.
- Звершення опрацювання теми: 03.11.2024 (15хв.).

### Виконання роботи:

1) *Опрацювання завдання та вимог до програми та середовища*

#### Завдання №1 VNS Lab 2 - Task 1-10

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

$$a_n = \frac{n!}{(2n)!}$$

1. При визначенні суми членів ряду варто використовувати рекурентну формулу для отримання наступного члена ряду.
2. При складанні програми вважати, що точність досягнута, якщо  $a_n < \epsilon$ .

#### Завдання №2 VNS Lab 3 - Task 1-10

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  $\epsilon$  ( $\epsilon=0.0001$ ).

Для порівняння знайти точне значення функції.

10	$y = e^{\cos x} \cos(\sin x)$	$0.1 \leq x \leq 1$	20	$S = 1 + \frac{\cos x}{1!} + \dots + \frac{\cos nx}{n!}$
----	-------------------------------	---------------------	----	--

1. Алгоритм розв'язання завдання зводиться до трьох циклів, причому два з них вкладені в третій. Внутрішні цикли підсумують доданки при фіксованому параметрі  $x$ , один (арифметичний для заданого  $n$ ), інший (ітераційний для заданої точності. При організації цих циклів варто

звернути увагу на правильний вибір формули для обчислення елемента ряду  $a_n$  і правильне присвоєння початкових значень змінним циклу. Зовнішній цикл організує зміну параметра  $x$ .

2. Результати розрахунків надрукувати у такому вигляді:

Обчислення функції			
X=.....	SN=.....	SE=.....	Y=.....
X=.....	SN=.....	SE=.....	Y=.....
.....			
X=.....	SN=.....	SE=.....	Y=.....

Тут X- значення параметра; SN- значення суми для заданого  $n$ ; SE- значення суми для заданої точності; Y-точне значення функції.

### Завдання №3 VNS Lab 7 - Task 1-10

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

Написати функцію `kvaдр` зі змінною кількістю параметрів, що визначає кількість чисел, що є точними квадратами (2, 4, 9, 16, . . .) типу `int`. Написати викликаючу функцію `main`, що звертається до функції `kvaдр` не менше трьох разів з кількістю параметрів 3, 7, 11.

### Завдання №4 VNS Lab 7 - Task 1-10

Написати перевантажені функції й основну програму, що їх викликає.

- а) для віднімання десяткових дробів;
- б) для віднімання звичайних дробів.

### 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**

**Загадкове число**

*Обмеження: 2 сек., 256 MiB*

Марічка, як і годиться усім представницям прекрасної статі, полюбляє говорити загадками. Так і цього разу, Марічка планувала написати Зенику записку з її улюбленим числом, проте вона не втрималася, і деякі цифри в числі замінила на зірочки (символ \*).

Коли Зеник отримав записку, він, як завжди, не зрозумів, що це мало означати і який прихований зміст у цьому повідомленні. Тому він вирішив визначити, яке мінімальне, та яке максимальне можливе число Марічка хотіла йому сказати.

Зверніть увагу, кожен зірочку треба замінити на якусь цифру. Також, числа не повинні містити ведучих нулів. Тобто, першою цифрою числа повинен бути не 0.

### **Вхідні дані**

У єдиному рядку задано один рядок — послідовність з цифр і зірочок.

### **Вихідні дані**

У єдиному рядку виведіть два натуральних числа — мінімальне та максимальне можливе число, яке Марічка хотіла повідомити.

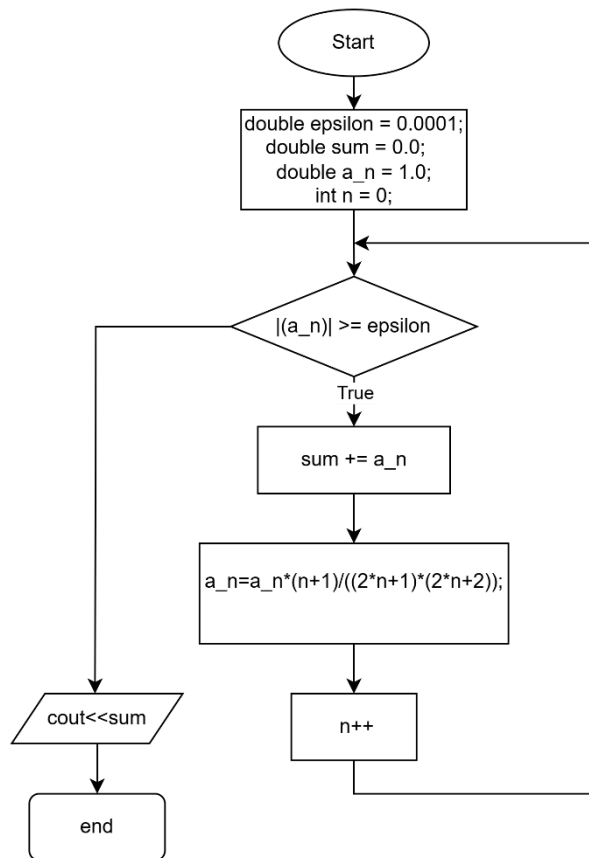
### **Обмеження**

Довжина числа не перевищує 9,  
гарантується, що перший символ не 0.

*2) Дизайн та планована оцінка часу виконання завдань:*

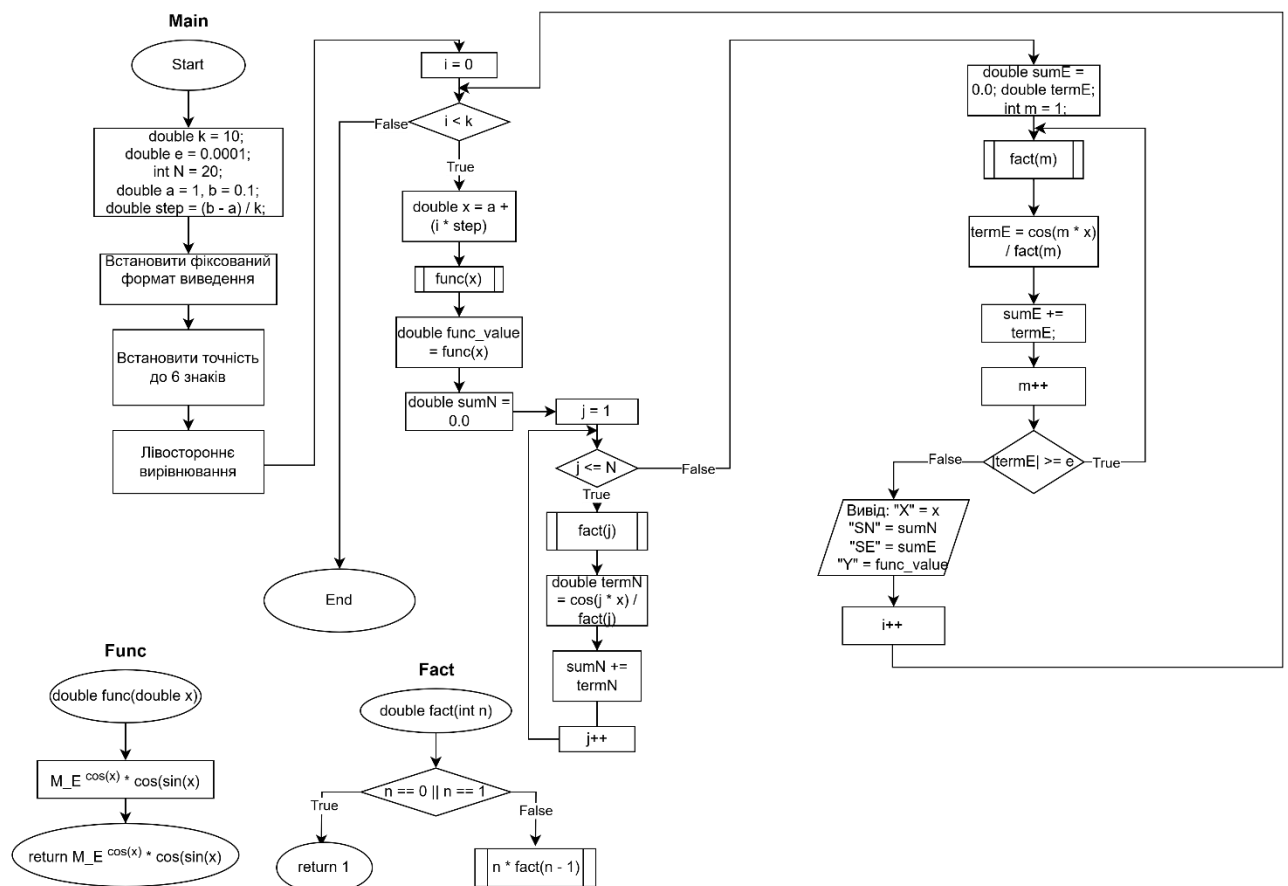
**Завдання №1 VNS Lab 2 - Task 1-10**





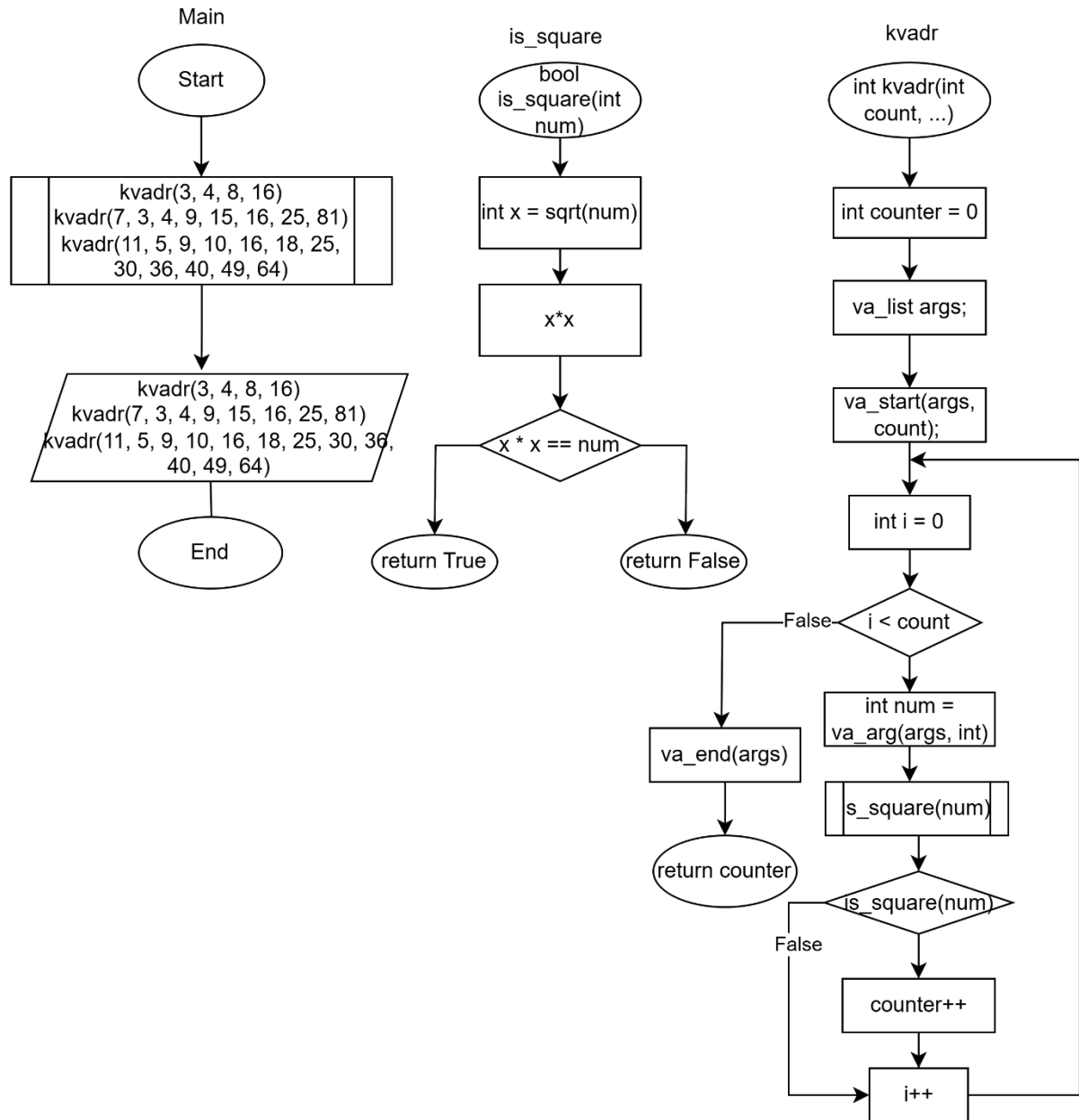
Планований час: 30 хв.

## Завдання №2 VNS Lab 3 - Task 1-10



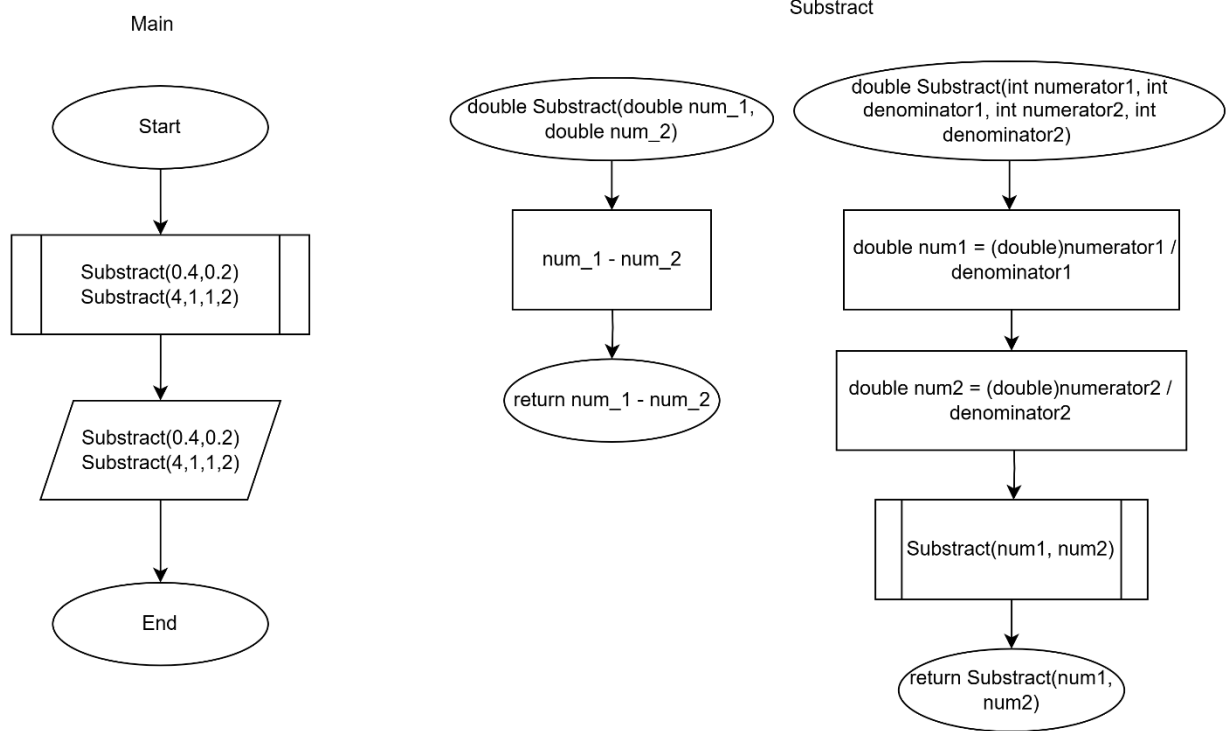
Планований час: 45 хв.

### Завдання №3 VNS Lab 7 - Task 1-10



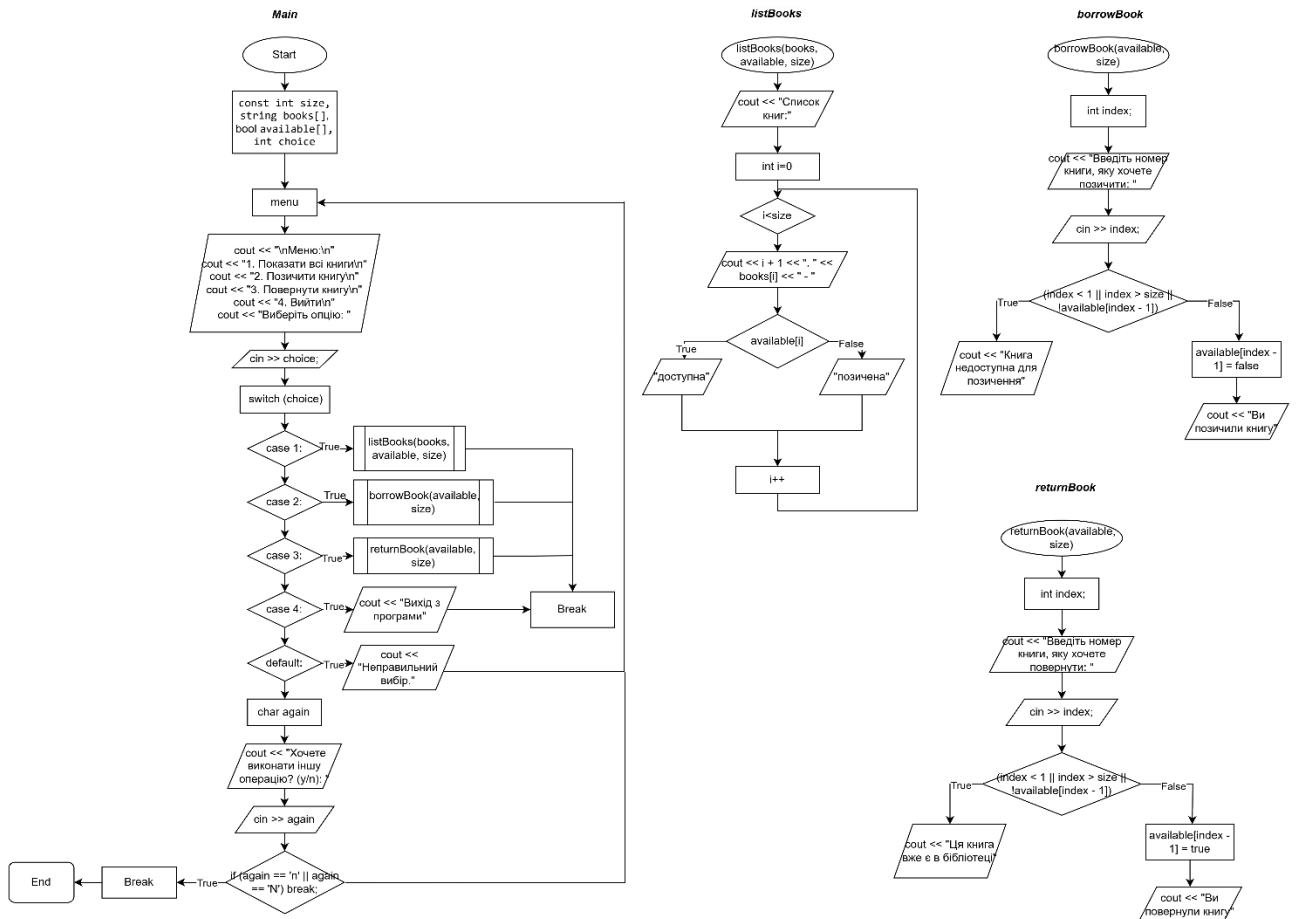
Планований час: 40 хв.

### Завдання №4 VNS Lab 7 - Task 1-10



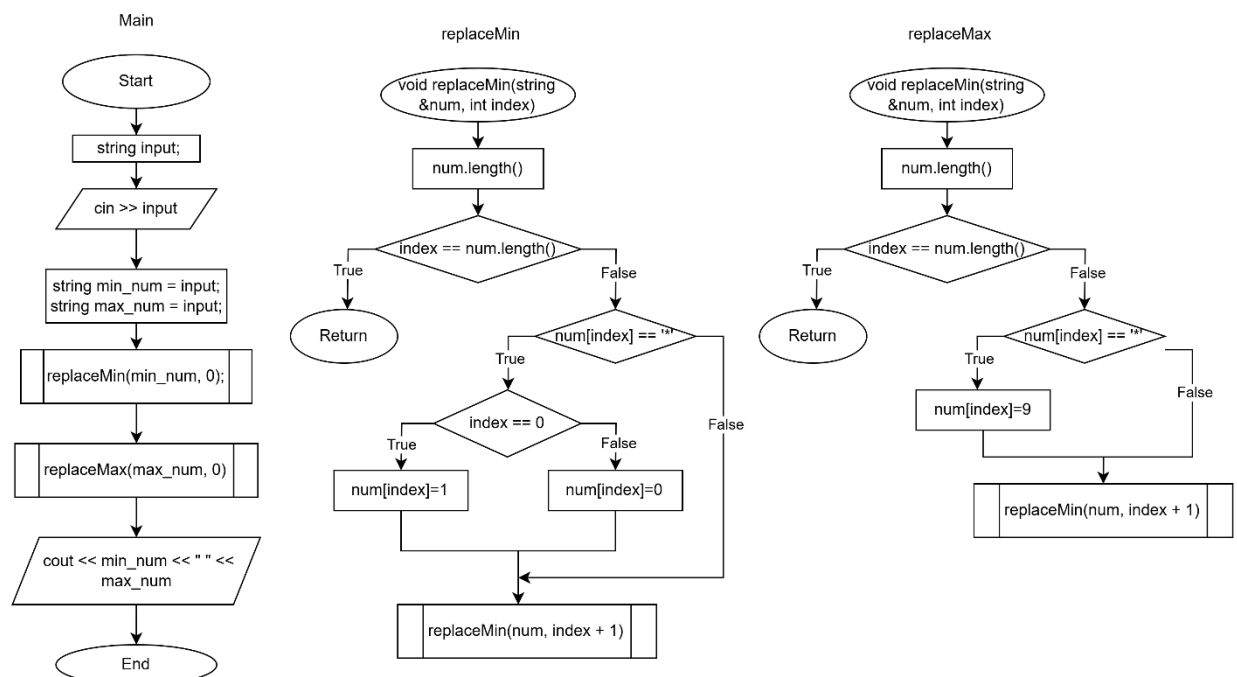
Планований час: 20 хв.

## Завдання №5 Class Practice Work



Планований час: 1 год.

## Завдання №6 Self Practice Work



Планований час: 30 хв.

### 3) Код програм з посиланням на зовнішні ресурси

#### Завдання №1

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    double epsilon = 0.0001;
    double sum = 0.0;
    double a_n = 1.0;
    int n = 0;

    while (abs(a_n) >= epsilon) {
        sum += a_n;

        a_n = a_n * (n + 1) / ((2 * n + 1) * (2 * n + 2));
        n++;
    }

    cout << "Сума ряду з точністю " << epsilon << " дорівнює: " << sum << endl;
    return 0;
}
```

#### Завдання №2

```
#define _USE_MATH_DEFINES
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

double func(double x) {
    return pow(M_E, cos(x)) * cos(sin(x));
}

double fact(int n) {
    return (n == 0 || n == 1) ? 1 : n * fact(n - 1);
}

int main() {
    double k = 10;
    double e = 0.0001;
    int N = 20;
    double a = 1, b = 0.1;
```

```

double step = (b - a) / k;

cout << fixed << setprecision(6);
cout << setiosflags(ios::left);

for (int i = 0; i < k; i++) {
    double x = a + (i * step);
    double func_value = func(x);

    double sumN = 0.0;
    for (int j = 1; j <= N; j++) {
        double termN = cos(j * x) / fact(j);
        sumN += termN;
    }

    double sumE = 0.0;
    double termE;
    int m = 1;
    do {
        termE = cos(m * x) / fact(m);
        sumE += termE;
        m++;
    } while (abs(termE) >= e);

    // Виведення результатів
    cout << "X=" << setw(10) << x
        << " SN=" << setw(10) << sumN
        << " SE=" << setw(10) << sumE
        << " Y=" << setw(10) << func_value << "\n";
}

return 0;
}

```

### Завдання №3

```

#include <iostream>
#include <cmath>
#include <cstdint>

using namespace std;

bool is_square(int num) {
    int x = sqrt(num);
    return x * x == num;
}

int kvadr(int count, ...) {

```

```

    int counter = 0;
    va_list args;
    va_start(args, count);

    for (int i = 0; i < count; i++) {
        int num = va_arg(args, int);
        if (is_square(num)) {
            counter++;
        }
    }

    va_end(args);
    return counter;
}

int main() {
    cout << "Кількість точних квадратів для 3: " << kvadr(3, 4, 8, 16) << endl;
    cout << "Кількість точних квадратів для 7: " << kvadr(7, 3, 4, 9, 15, 16, 25,
81) << endl;
    cout << "Кількість точних квадратів для 11: " << kvadr(11, 5, 9, 10, 16, 18,
25, 30, 36, 40, 49, 64) << endl;

    return 0;
}

```

## Завдання №4

```

#include <iostream>
#include <cmath>
using namespace std;
double Subtract(double num_1, double num_2)
{
    return num_1 - num_2;
}

double Subtract(int numerator1, int denominator1, int numerator2, int
denominator2)
{
    double num1 = (double)numerator1 / denominator1;
    double num2 = (double)numerator2 / denominator2;
    return Subtract(num1, num2);
}

int main()
{
    cout << "Десяткові дробі: " << Subtract(0.4, 0.2) << endl;
    cout << "Звичайні дробі: " << Subtract(4, 1, 1, 2) << endl;

    return 0;
}

```

```
}
```

## Завдання №5

```
#include <iostream>
#include <string>
using namespace std;

void listBooks(const string books[], const bool available[], int size) {
    cout << "Список книг:" << endl;
    for (int i = 0; i < size; i++) {
        cout << i + 1 << ". " << books[i] << " - " << (available[i] ? "доступна"
: "позичена") << endl;
    }
}

void borrowBook(bool available[], int size) {
    int index;
    cout << "Введіть номер книги, яку хочете позичити: ";
    cin >> index;
    if (index < 1 || index > size || !available[index - 1]) {
        cout << "Книга недоступна для позичення" << endl;
    } else {
        available[index - 1] = false;
        cout << "Ви позичили книгу" << endl;
    }
}

void returnBook(bool available[], int size) {
    int index;
    cout << "Введіть номер книги, яку хочете повернути: ";
    cin >> index;
    if (index < 1 || index > size || available[index - 1]) {
        cout << "Ця книга вже є в бібліотеці" << endl;
    } else {
        available[index - 1] = true;
        cout << "Ви повернули книгу" << endl;
    }
}

int main() {
    const int size = 3;
    string books[size] = {"Книга 1", "Книга 2", "Книга 3"};
    bool available[size] = {true, true, true};
    int choice;

    do {
        menu:
        cout << "\nМеню:\n";
        cout << "1. Показати всі книги\n";
```



```

    cout << "2. Позичити книгу\n";
    cout << "3. Повернути книгу\n";
    cout << "4. Вийти\n";
    cout << "Виберіть опцію: ";
    cin >> choice;

    switch (choice) {
        case 1:
            listBooks(books, available, size);
            break;
        case 2:
            borrowBook(available, size);
            break;
        case 3:
            returnBook(available, size);
            break;
        case 4:
            cout << "Вихід з програми" << endl;
            break;
        default:
            cout << "Неправильний вибір." << endl;
            goto menu;
    }

    char again;
    cout << "Хочете виконати іншу операцію? (y/n): ";
    cin >> again;
    if (again == 'n' || again == 'N') break;

} while (true);

return 0;
}

```

## Завдання №6

```

#include <iostream>
#include <string>
using namespace std;

void replaceMin(string &num, int index) {
    if (index == num.length()) return;
    if (num[index] == '*') {
        num[index] = (index == 0) ? '1' : '0';
    }
    replaceMin(num, index + 1);
}

void replaceMax(string &num, int index) {

```

```

        if (index == num.length()) return;
        if (num[index] == '*') {
            num[index] = '9';
        }
        replaceMax(num, index + 1);
    }

int main() {
    string input;
    cin >> input;

    string min_num = input;
    string max_num = input;

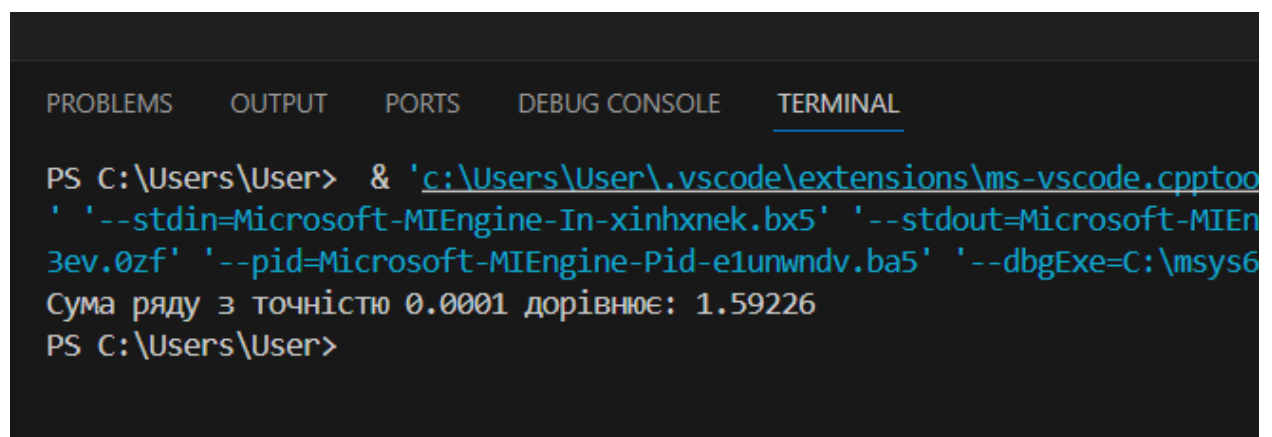
    replaceMin(min_num, 0);
    replaceMax(max_num, 0);

    cout << min_num << " " << max_num << endl;
    return 0;
}

```

4) *Результати виконання завдань, тестування та фактично затрачений час:*

#### **Завдання №1 VNS Lab 2 - Task 1-10**



```

PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

PS C:\Users\User> & 'c:\Users\User\.vscode\extensions\ms-vscode.cpptool
' '--stdin=Microsoft-MIEngine-In-xinhxnek.bx5' '--stdout=Microsoft-MIEn
Зев.0zf' '--pid=Microsoft-MIEngine-Pid-e1unwndv.ba5' '--dbgExe=C:\msys6
Сума ряду з точністю 0.0001 дорівнює: 1.59226
PS C:\Users\User>

```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_3\\_practice\\_and\\_labs\\_anastasiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic3/vns\\_lab\\_2\\_task\\_1\\_variant\\_10\\_anastasiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic3/vns_lab_2_task_1_variant_10_anastasiia_humeniuk.cpp)

Фактично затрачений час: 35 хв.

#### **Завдання №2 VNS Lab 3 - Task 1-10**

```
PROBLEMS    OUTPUT    PORTS    DEBUG CONSOLE    TERMINAL

X=1.000000    SN=0.143836    SE=0.143838    Y=1.143836
X=0.910000    SN=0.300891    SE=0.300893    Y=1.300891
X=0.820000    SN=0.472641    SE=0.472640    Y=1.472641
X=0.730000    SN=0.655450    SE=0.655425    Y=1.655450
X=0.640000    SN=0.844196    SE=0.844184    Y=1.844196
X=0.550000    SN=1.032383    SE=1.032383    Y=2.032383
X=0.460000    SN=1.212426    SE=1.212427    Y=2.212426
X=0.370000    SN=1.376117    SE=1.376120    Y=2.376117
X=0.280000    SN=1.515252    SE=1.515270    Y=2.515252
X=0.190000    SN=1.622330    SE=1.622330    Y=2.622330
PS C:\Users\User>
```

Фактично затрачений час: 1 год.

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_3\\_practice\\_and\\_labs\\_anastasiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic3/vns\\_lab\\_3\\_task\\_1\\_variant\\_10\\_anastasiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic3/vns_lab_3_task_1_variant_10_anastasiia_humeniuk.cpp)

### Завдання №3 VNS Lab 7 - Task 1-10

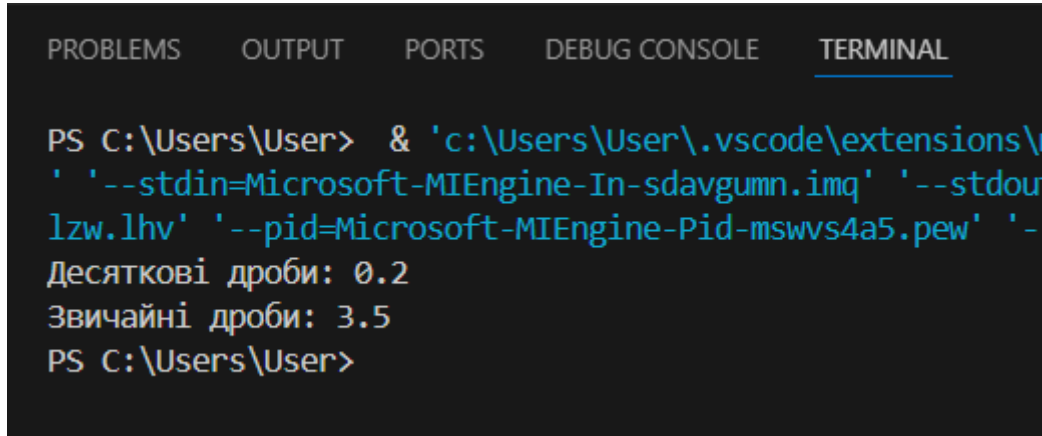
```
PROBLEMS    OUTPUT    PORTS    DEBUG CONSOLE    TERMINAL

PS C:\Users\User> & 'c:\Users\User\.vscode\extensions\ms
' '--stdin=Microsoft-MIEngine-In-kuhsr0pj.w41' '--stdout=
zfl.ih5' '--pid=Microsoft-MIEngine-Pid-uzdbqzey.00j' '--c
Кількість точних квадратів для 3: 2
Кількість точних квадратів для 7: 5
Кількість точних квадратів для 11: 6
PS C:\Users\User>
```

Фактично затрачений час: 30 хв.

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_3\\_practice\\_and\\_labs\\_anastasiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic3/vns\\_lab\\_7\\_task\\_1\\_variant\\_10\\_anastasiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic3/vns_lab_7_task_1_variant_10_anastasiia_humeniuk.cpp)

## Завдання №4 VNS Lab 7 - Task 2-10



The screenshot shows a VS Code interface with a terminal window open. The terminal has tabs for PROBLEMS, OUTPUT, PORTS, DEBUG CONSOLE, and TERMINAL. The terminal content shows a PowerShell command being executed in the C:\Users\User directory. The command is: `& 'c:\Users\User\.vscode\extensions\ms-vscode-ai\ms-vscode-ai\cli\cli.ps1' --stdin=Microsoft-MIEngine-In-sdavgumn.img --stdout=Microsoft-MIEngine-Out-sdavgumn.lzw.lhv --pid=Microsoft-MIEngine-Pid-mswvs4a5.pew --`. The output of the command is: `Десяткові дробі: 0.2` and `Звичайні дробі: 3.5`. The terminal prompt is `PS C:\Users\User>`.

```
PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

PS C:\Users\User> & 'c:\Users\User\.vscode\extensions\ms-vscode-ai\ms-vscode-ai\cli\cli.ps1' --stdin=Microsoft-MIEngine-In-sdavgumn.img --stdout=Microsoft-MIEngine-Out-sdavgumn.lzw.lhv --pid=Microsoft-MIEngine-Pid-mswvs4a5.pew --
Десяткові дробі: 0.2
Звичайні дробі: 3.5
PS C:\Users\User>
```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_3\\_practice\\_and\\_labs\\_anastasiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic3/vns\\_lab\\_7\\_task\\_2\\_variant\\_10\\_anastasiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic3/vns_lab_7_task_2_variant_10_anastasiia_humeniuk.cpp)

Фактично затрачений час: 20 хв.

## Завдання №5 Class Practice Work

```
xzk.zmp' '--pid=Microsoft-MIEngine-Pid-hjbyom4b.d3s' '--d

Меню:
1. Показати всі книги
2. Позичити книгу
3. Повернути книгу
4. Вийти
Виберіть опцію: 2
Введіть номер книги, яку хочете позичити: 1
Ви позичили книгу
Хочете виконати іншу операцію? (y/n): y

Меню:
1. Показати всі книги
2. Позичити книгу
3. Повернути книгу
4. Вийти
Виберіть опцію: 1
Список книг:
1. Книга 1 - позичена
2. Книга 2 - доступна
3. Книга 3 - доступна
Хочете виконати іншу операцію? (y/n): y

Меню:
1. Показати всі книги
2. Позичити книгу
3. Повернути книгу
4. Вийти
Виберіть опцію: 3
Введіть номер книги, яку хочете повернути: 1
Ви повернули книгу
Хочете виконати іншу операцію? (y/n): y

Меню:
1. Показати всі книги
2. Позичити книгу
3. Повернути книгу
4. Вийти
Виберіть опцію: 3
Введіть номер книги, яку хочете повернути: 1
Ця книга вже є в бібліотеці
Хочете виконати іншу операцію? (y/n): n
PS C:\Users\User> █
```

Фактично затрачений час: 50 хв.

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_3\\_practice\\_and\\_labs\\_anastasiiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic3/practice\\_work\\_task\\_1\\_anastasiiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_anastasiiia_humeniuk/ai_11/anastasiia_humeniuk/epic3/practice_work_task_1_anastasiiia_humeniuk.cpp)

**Завдання №6 Self Practice Work**

```
PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

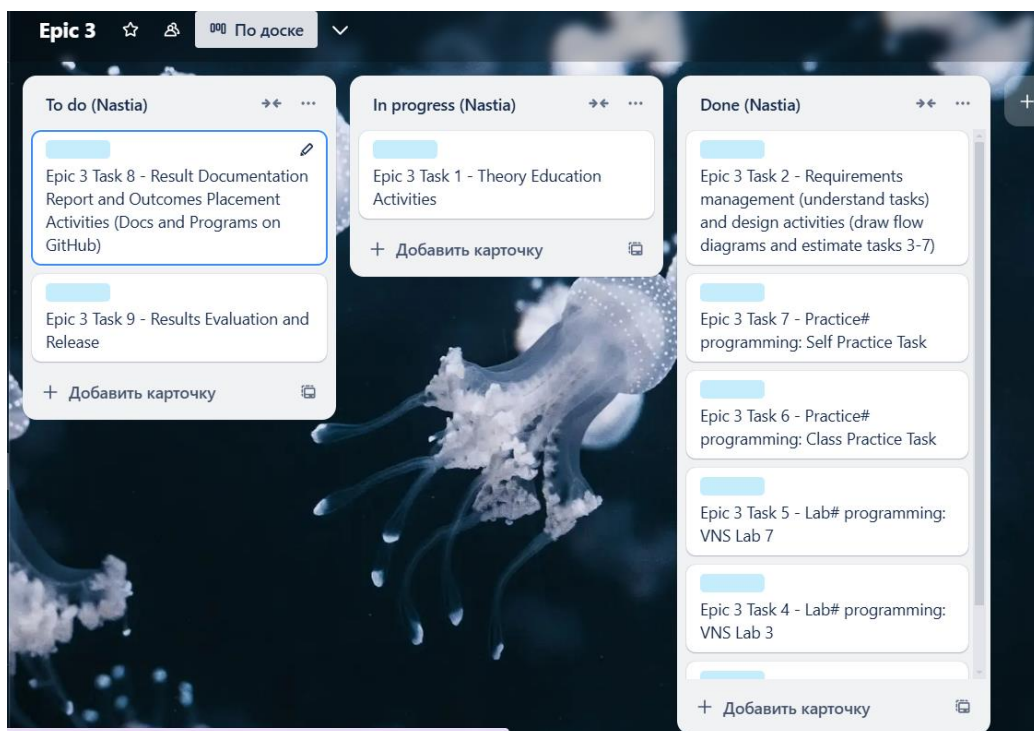
PS C:\Users\User> & 'c:\Users\User\.vscode\extensions\ms-vscode.c
' '--stdin=Microsoft-MIEngine-In-jjiqhew1.qnb' '--stdout=Microsoft
1no.cqc' '--pid=Microsoft-MIEngine-Pid-ylInftpe.e3o' '--dbgExe=C:\
*56*2**
1560200 9569299
PS C:\Users\User> ^C
PS C:\Users\User>
PS C:\Users\User> & 'c:\Users\User\.vscode\extensions\ms-vscode.c
' '--stdin=Microsoft-MIEngine-In-0qxdj3u0.jfb' '--stdout=Microsoft
du4.5nx' '--pid=Microsoft-MIEngine-Pid-kq1doprz.kz4' '--dbgExe=C:\
47
47 47
PS C:\Users\User> |
```

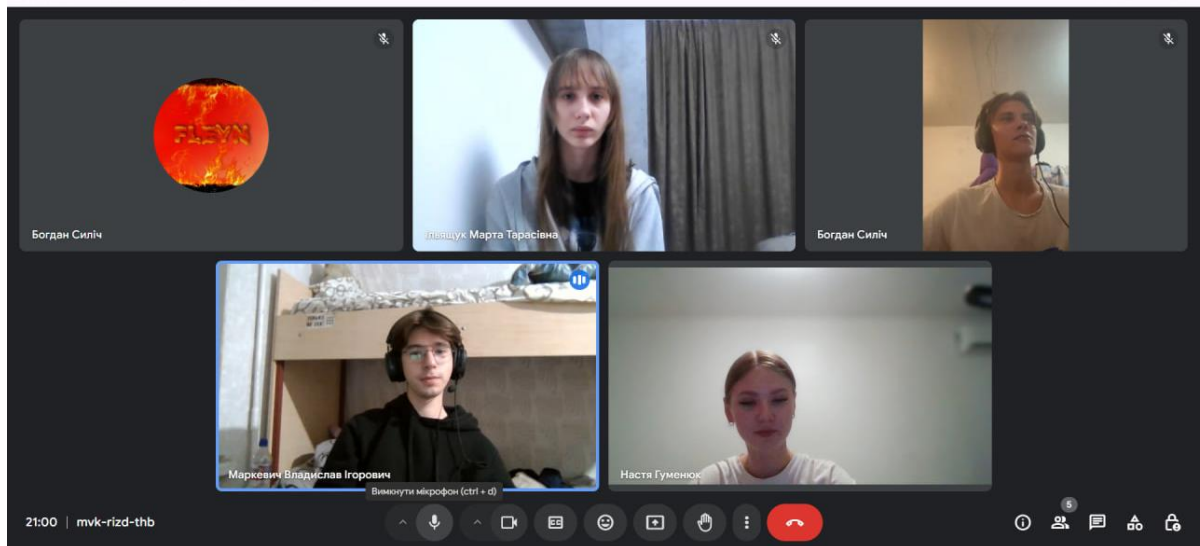
Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
3 години тому	C++ 23	Зараховано	0.002	1.207	Перегляд

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_3\\_practice\\_and\\_labs\\_anastasiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic3/self\\_practice\\_work\\_algotester\\_task\\_1\\_anastasiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic3/self_practice_work_algotester_task_1_anastasiia_humeniuk.cpp)

Фактично затрачений час: 30 хв.

5) Кооперація з командою:





**Висновок:** Виконуючи третій епік я ознайомила з різними видами циклів у C++ і зрозуміла, як вони можуть спростити реалізацію повторюваних завдань у програмуванні. Навчилася керувати виконанням циклів за допомогою операторів `break` і `continue`, що дозволяє змінювати логіку їх виконання. Дослідження функцій, їх параметрів і перевантаження допомогло мені краще організовувати код і робити його більш зрозумілим. Я також опанувала застосування рекурсії та функцій зі змінною кількістю параметрів, що розширило мої знання про можливості функцій у складних завданнях. Завдяки вивченню вбудованих функцій, я побачила, як C++ може оптимізувати виконання стандартних операцій і полегшити написання коду. А також на практиці відпрацювала ці знання, написавши декілька програм.