

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:

Студент групи ШІ-12

Михальчук Антон Євгенійович

Тема роботи:

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

Мета роботи:

Ознайомитися з основними концепціями роботи з циклами, включаючи вкладені цикли та методи завершення їх виконання. Навчитися використовувати функції в програмуванні, розглянути поняття простору імен та зрозуміти принципи перевантаження функцій для підтримки різних типів аргументів. Ознайомитися з використанням функцій зі змінною кількістю параметрів (еліпсис), а також із поняттям рекурсії. Розглянути використання вбудованих функцій для спрощення розв'язання завдань та їх застосування в програмному коді.

Теоретичні відомості:

- 1) Теоретичні відомості з переліком важливих тем:
 - Тема №*.1: C++ Cycles, functions
- 2) Індивідуальний план опрацювання теорії:
 - Тема №*.1: C++ Basics
 - o Джерела Інформації
 - Відео. <https://www.youtube.com/watch?v=2UDMGCCcRCjo>
 - Стаття. <https://www.w3schools.com/cpp/>
 -
 - o Що опрацьовано:
 - Вивчив базовий синтаксис та семантику мови C++.
 - Особливу увагу приділяв таким темам, як цикли та функції
 - o Статус: Ознайомлений
 - o Початок опрацювання теми: 15.09.2024
 - o Звершення опрацювання теми: 6.11.2024

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Варіант 9

Завдання №1 VNS Lab 2

- Деталі завдання:
 - o Використовуючи оператор циклу, знайти суму елементів, зазначених у конкретному варіанті. Результат надрукувати, надавши відповідний заголовком.
Знайти суму ряду з точністю $\varepsilon=0.0001$, загальний член якого:
- Важливі деталі для врахування в імплементації програми:
 - o При визначенні суми членів ряду варто використовувати рекурентну формулу для отримання наступного члена ряду.

$$a_n = \frac{10^n}{n!}$$

- o При складанні програми вважати, що точність досягнута, якщо $a_n < \epsilon$

Завдання №2 VNS Lab 3

- Деталі завдання:
 - o Для x , що змінюється від a до b з кроком $(b-a)/k$, де $(k=10)$, обчислити
 - o функцію $f(x)$, використовуючи її розклад в степеневий ряд у двох випадках:
 - а) для заданого n ;
 - б) для заданої точності ϵ ($\epsilon=0.0001$).
 - o Для порівняння знайти точне значення функції.
- Важливі деталі для врахування в імплементації програми:
 - o 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 ϵ).
 - o 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 .

Завдання №3 VNS Lab 7 Task 1

- Деталі завдання:
 - o Написати функцію `min` зі змінною кількістю параметрів, що знаходить мінімальне із чисел типу `int` або із чисел типу `double`, тип параметрів визначається за допомогою першого параметра функції.
 - o Написати викликаючу функцію `main`, що звертається до функції `min` не менше трьох разів з кількістю параметрів 5, 10, 12.
- Важливі деталі для врахування в імплементації програми:
 - o Методичні вказівки мають не правильний код, тому варто використовувати шаблони.

Завдання №4 VNS Lab 7 Task 2

- Деталі завдання:
 - o Написати перевантажені функції й основну програму, що їх викликає.
 - а) для додавання десяткових дробів;
 - б) для додавання звичайних дробів.
- Важливі деталі для врахування в імплементації програми:
 - o Типами для десяткових дробів будуть `double` та `float`.
 - o Типами для звичайних дробів будуть пари з `int` та `long long`.

Завдання №5 Class Practice Task

- Деталі завдання:
 - o Ви створюєте просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.
 - o Програма повинна вміти
 - Перерахувати всі книги.
 - Дозволити взяти книгу (за наявності).
 - Дозволити повернення книги.
- Важливі деталі для врахування в імплементації програми:
 - o `while`: продовжувати працювати, доки користувач не вирішить вийти.

- o do while: Після кожної операції (позичити, повернути, перерахувати) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.
- o for: список усіх книг за допомогою циклу.
- o for each: перевірити наявність кожної книги.
- o goto: якщо користувач вводить неправильний вибір, використовуйте goto, щоб перенаправити його до головного меню.

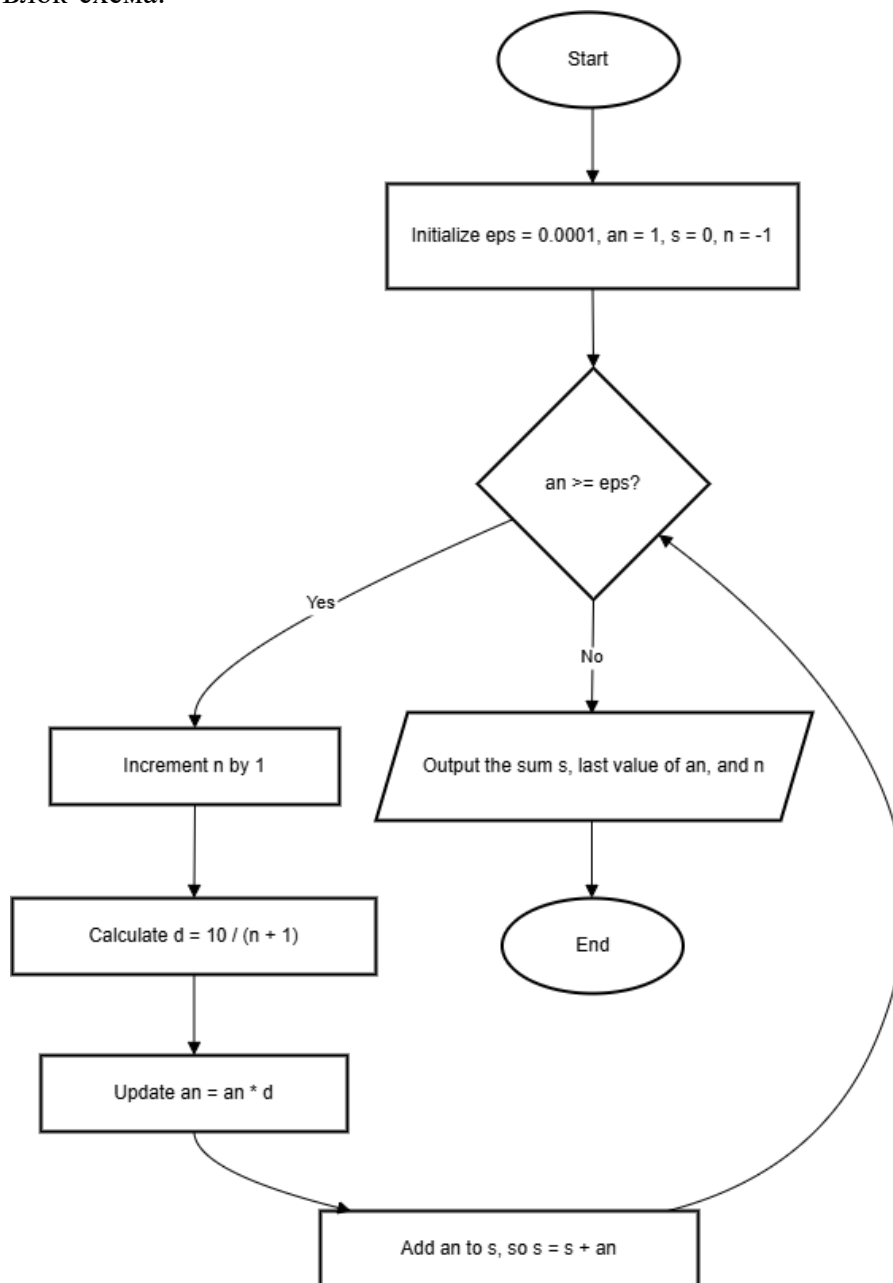
Завдання №6 Self Practice Task Коля, Вася і Теніс

- Деталі завдання:
 - o Всього відбулося N подач. Про кожен з них ми знаємо, хто переміг. За виграну подачу гравець отримує одне очко. Партія вважається виграною, коли один з гравців набере не менше одинадцяти очок з перевагою щонайменше у два очки. Наприклад, за рахунків 11:9, 4:11, 15:13 партія закінчується, а за рахунків 11:10 та 99:98 — ні. Як тільки Коля і Вася закінчили одну партію — вони починають іншу.
 - o Знаючи, хто переміг кожної подачі — виведіть загальний рахунок по партіях в грі Коля-Вася. А якщо вони не дограли останню партію, то і її рахунок теж.
- Важливі деталі для врахування в імплементації програми:
 - o У першому рядку задано ціле число n — загальна кількість подач.
 - o У другому рядку задано n символів: K, якщо i-ту подачу виграв Коля, та V, якщо i-ту подачу виграв Вася.
 - o У першому рядку виведіть загальний рахунок гри по партіях у форматі k:v, де k — кількість партій, у яких переміг Коля, а v — кількість партій, у яких переміг Вася.
 - o Якщо вони не дограли останню партію, то в другому рядку в такому ж форматі виведіть рахунок останньої партії.

2. Дизайн та планована оцінка часу виконання завдань:

Програма №1 VNS Lab 2

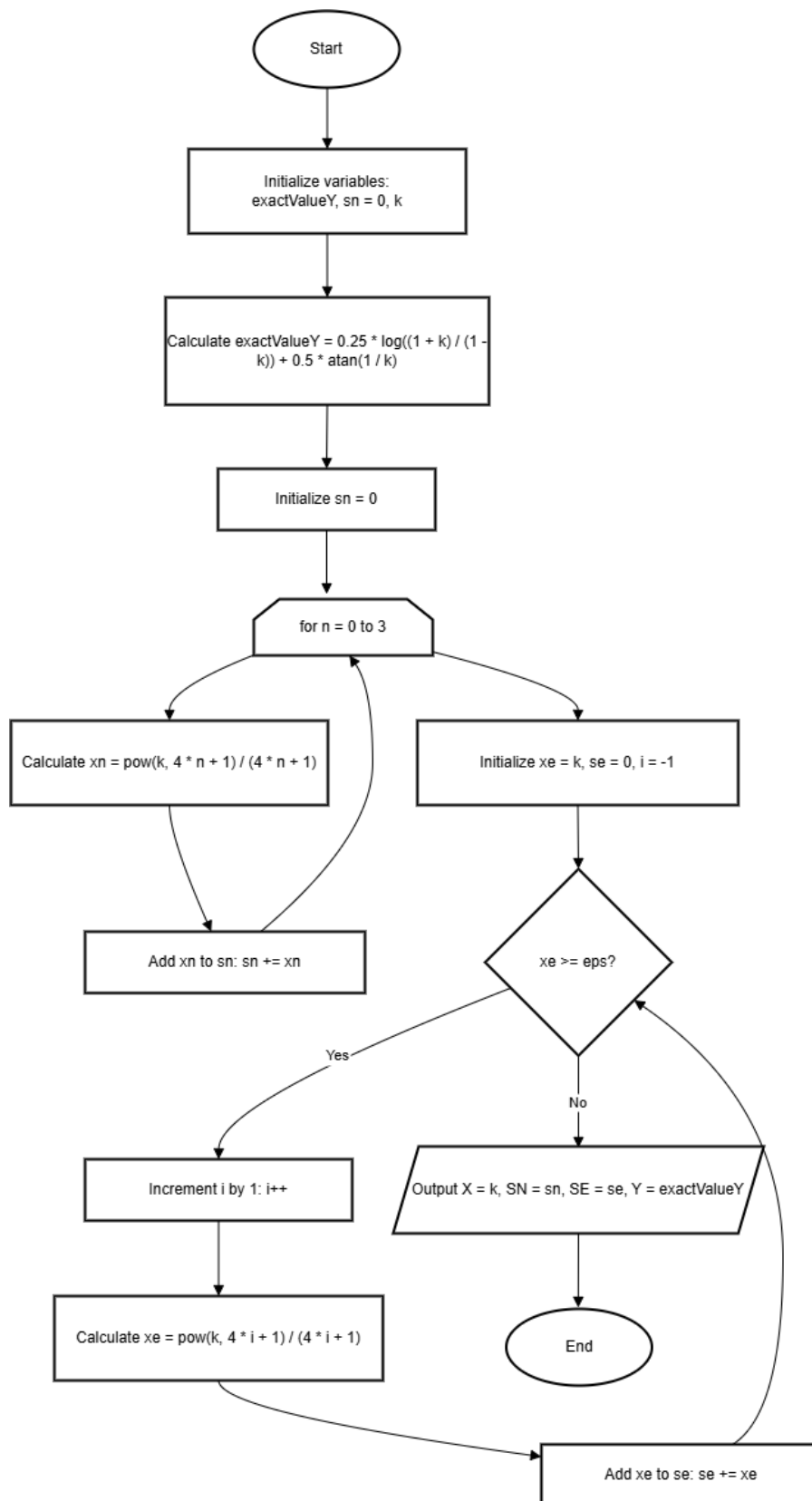
- Блок-схема:



- Планований час на реалізацію: 40 хв

Програма №2 VNS Lab 3

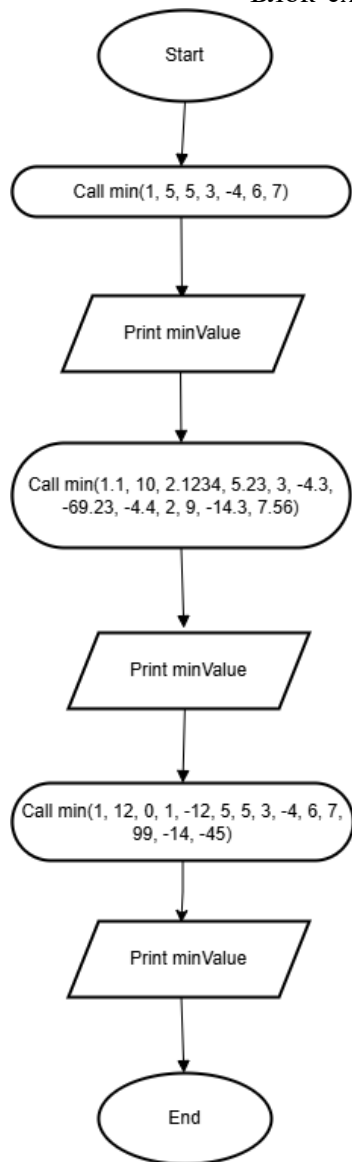
- Блок-схема:



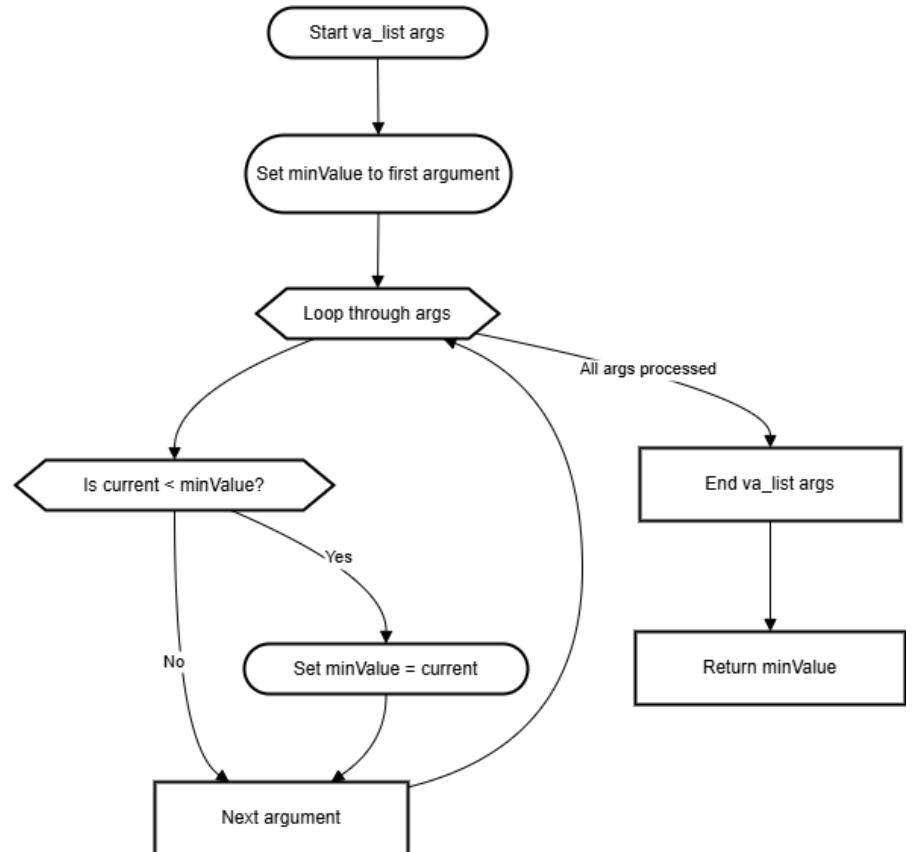
- Планований час на реалізацію: 40 хв.

Програма №3 VNS Lab 7 Task 1

- Блок-схема



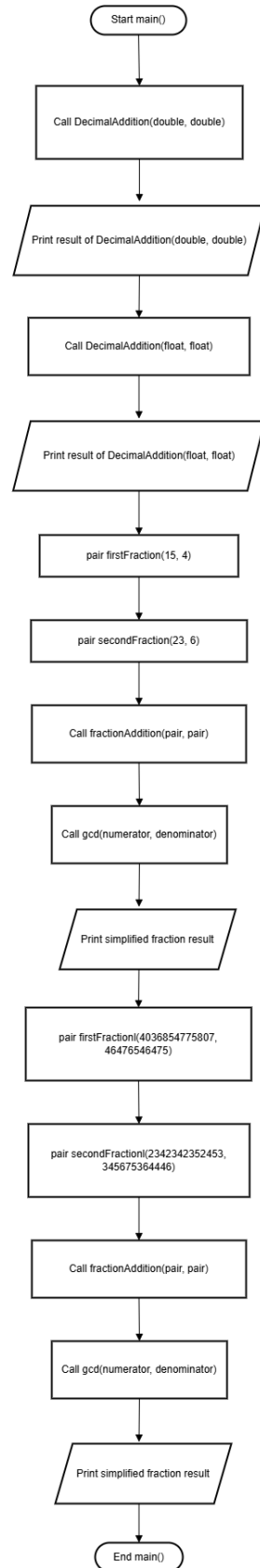
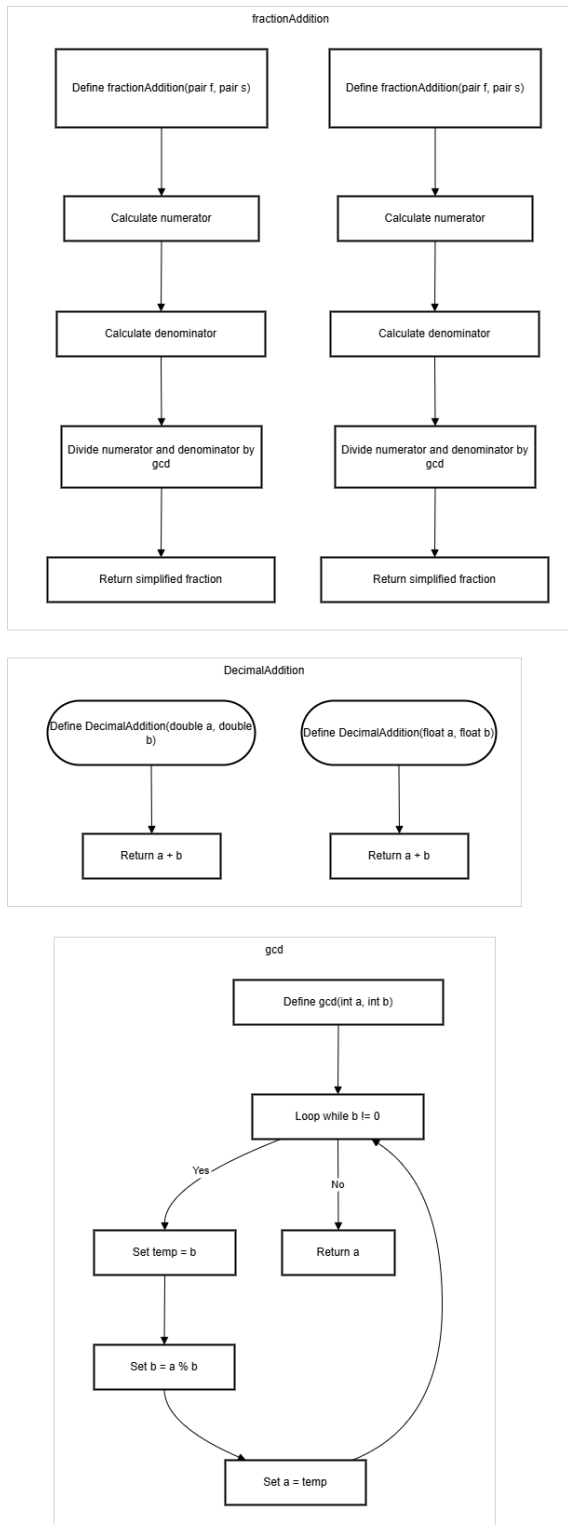
min() Function



- Планований час на реалізацію: 50 хв.

Програма №4 VNS Lab 7 Task 2

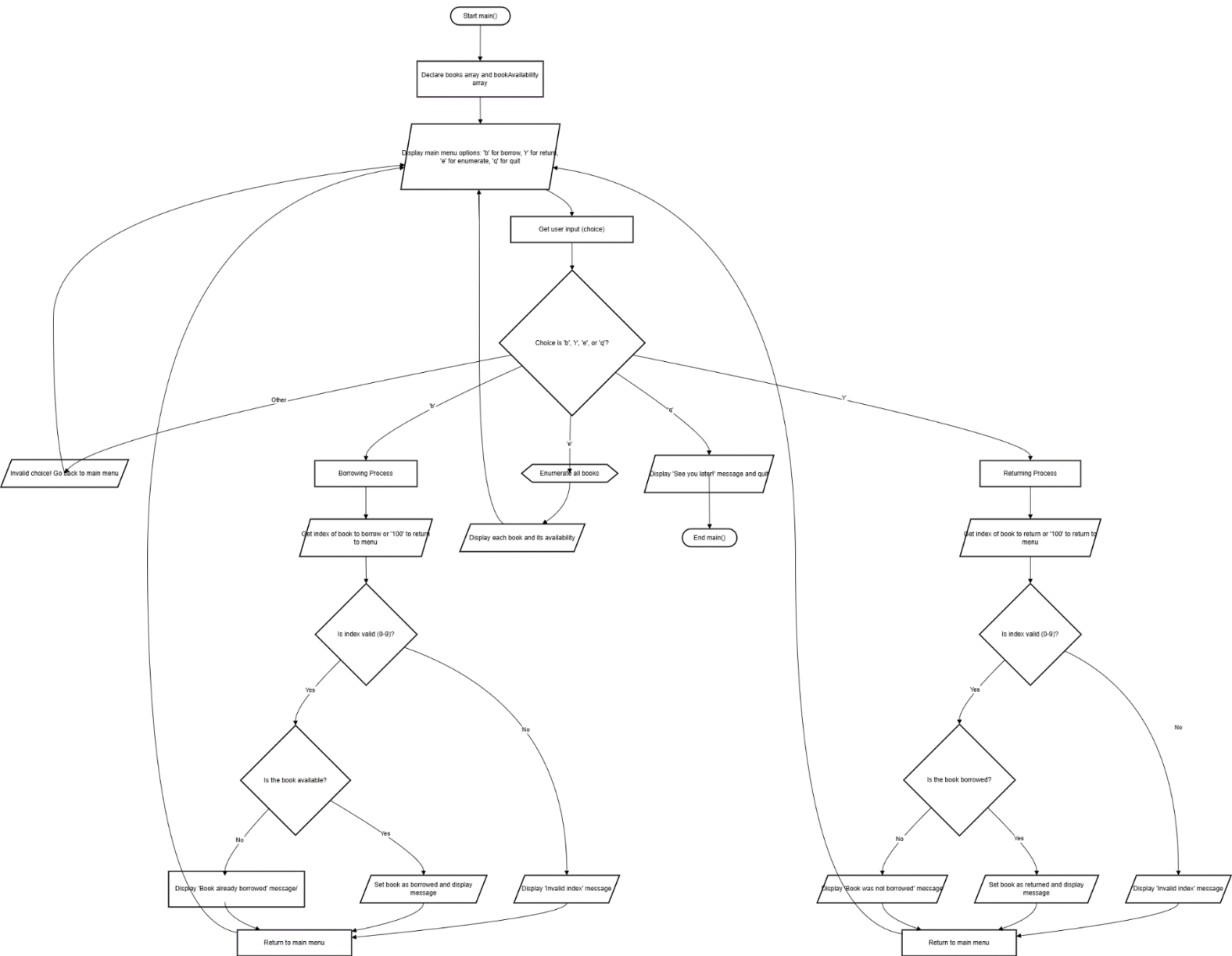
- Блок-схема



- Планований час на реалізацію: 1 год.

Програма №5 Class Practice Task

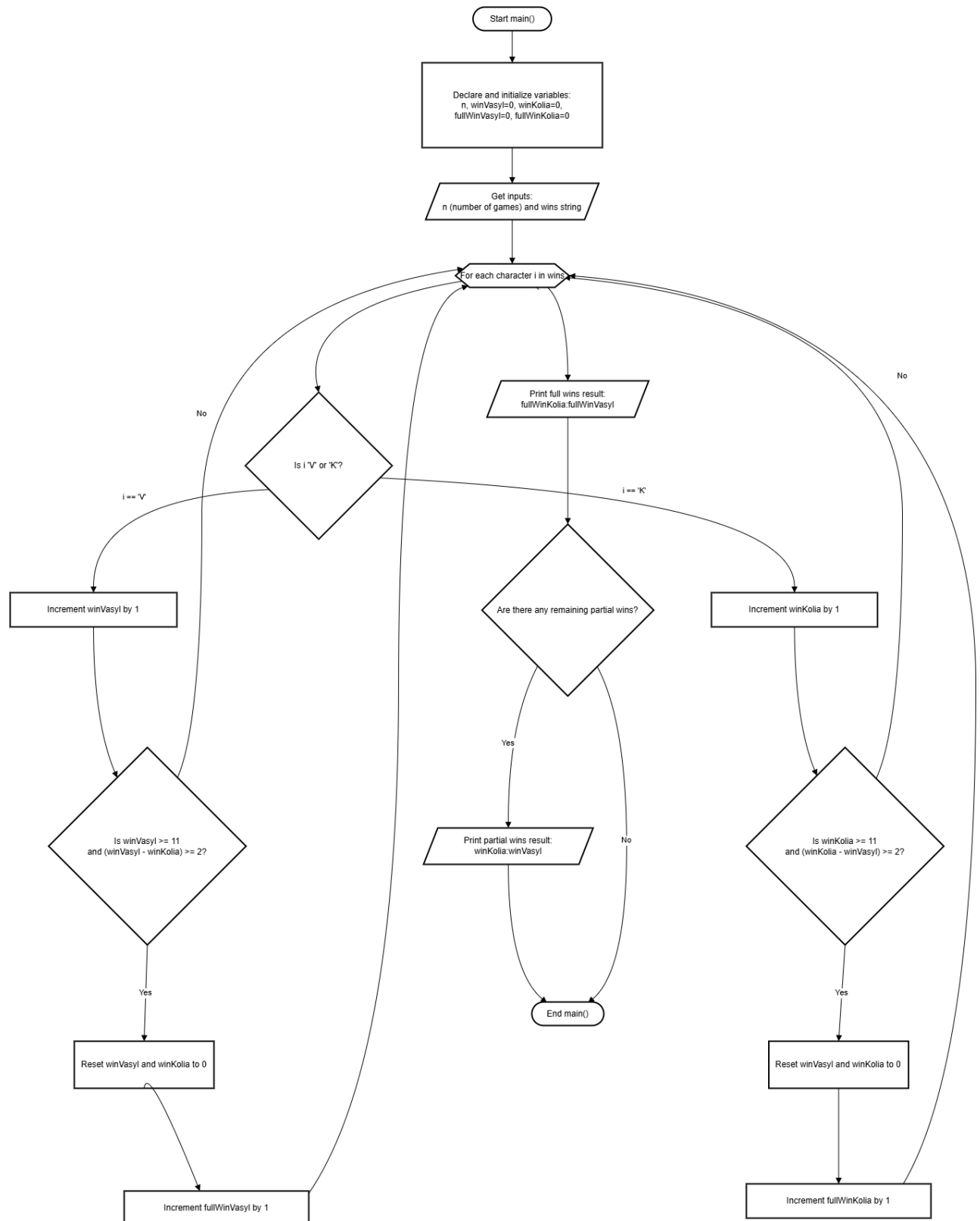
- Блок-схема



- Планований час на реалізацію: 50 хв.

Програма №6 Self Practice Task Коля, Вася і Теніс

- Блок-схема



- Планований час на реалізацію: 50 хв

4. Код програм з посиланням на зовнішні ресурси:

Завдання №1

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-6c05650ce74c77e06758635acd9b094333fca212b26733039bb5a3e757b76535

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

int main()
{
    const double eps = 0.0001;
    double an = 1, s = 0, d, n = -1;

    while (an >= eps)
    {
        n++;
        d = 10 / (n + 1);
        an *= d;
        s += an;
    }

    cout << "The sum of the series with an accuracy of epsilon = " << eps << "
is " << s << " The value is reached at an = " << an << " when n = " << n;
    return 0;
}
```

Завдання №2

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-415abdda3dfca6073b5be835e7de758b17dd9d99c8d2d62ef64948fc8359bfcf

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

int main()
{
    const double eps = 0.0001;
    double exactValueY, xn, sn, xe, se;
    int i;

    for (double k = 0.1; k <= 0.8 + eps; k += 0.07)
    {
        exactValueY = 0.25 * log(((1 + k) / (1 - k))) + 0.5 * atan(1 / k);
        sn = 0;
    }
}
```

```

    for (int n = 0; n <= 3; n++)
    {
        xn = pow(k, (4 * n + 1)) / (4 * n + 1);
        sn += xn;
    }
    xe = k;
    se = 0;
    i = -1;
    while (xe >= eps)
    {
        i++;
        xe = pow(k, (4 * i + 1)) / (4 * i + 1);
        se += xe;
    }

    cout << "X = " << k << " SN = " << sn << " SE = " << se << " Y = " <<
exactValueY << endl;
}

return 0;
}

```

Завдання №3

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-fd57a95df204ab21a978c82bcea0552fcac383c5f6758a63ac5801542b6ba880

```

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

template<typename T>
T min(T argType, int count, ...)
{
    va_list args;
    va_start(args, count);
    T minValue = va_arg(args, T);

    for (int i = 1; i < count; i++)
    {
        T current = va_arg(args, T);
        if (current < minValue)
        {
            minValue = current;
        }
    }
}

```

```

    }

    va_end(args);
    return minValue;
}

int main()
{
    cout << min(1, 5, 5, 3, -4, 6, 7) << endl;
    cout << min(1.1, 10, 2.1234, 5.23, 3, -4.3, -69.23, -4.4, 2, 9, -14.3,
7.56) << endl;
    cout << min(1, 12, 0, 1, -12, 5, 5, 3, -4, 6, 7, 99, -14, -45) << endl;

    return 0;
}

```

Завдання №4

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-bf12ec01580da6ffba449f66411411c1665a5ce581376ca108c9177703a70581

```

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

double DecimalAddition(double a, double b)
{
    return a + b;
}

float DecimalAddition(float a, float b)
{
    return a + b;
}

int gcd(int a, int b)
{
    while (b != 0)
    {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

```

```

pair<int, int> fractionAddition(pair<int, int> f, pair<int, int> s)
{
    int numerator = f.first * s.second + s.first * f.second;
    int denominator = f.second * s.second;
    int d = gcd(numerator, denominator);
    numerator /= d;
    denominator /= d;

    return make_pair(numerator, denominator);
}

pair<long long, long long> fractionAddition(pair<long long, long long> f,
pair<long long, long long> s)
{
    long long numerator = f.first * s.second + s.first * f.second;
    long long denominator = f.second * s.second;
    long long d = gcd(numerator, denominator);
    numerator /= d;
    denominator /= d;

    return make_pair(numerator, denominator);
}

int main()
{
    double ad = 45.434, bd = 2234.345;
    float af = 45.434, bf = 2234.345;

    cout << DecimalAddition(ad, bd) << endl;
    cout << DecimalAddition(af, bf) << endl;

    pair<int, int> firstFraction(15, 4), secondFraction(23, 6);
    pair<int, int> result = fractionAddition(firstFraction, secondFraction);
    cout << result.first << "/" << result.second << endl;

    pair<long long, long long> firstFractionl(4036854775807, 46476546475),
secondFractionl(2342342352453, 345675364446);
    pair<long long, long long> resultl = fractionAddition(firstFractionl,
secondFractionl);
    cout << resultl.first << "/" << resultl.second << endl;
    return 0;
}

```

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-85cddba83a0396e0ccc52b78ea051c8e0d61827275a4573407dcf3b40ac618a7

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

int main()
{
    char choice;
    string books[10] = {"1984", "To Kill a Mockingbird", "The Great Gatsby",
        "Moby-Dick", "Pride and Prejudice",
        "The Catcher in the Rye", "Brave New World", "Kobzar",
        "Taras Bulba", "The Hobbit"};
    bool bookAvailability[10] = {true, true, true, true, true, true, true,
        true, true, true};
    int borr, ret;
    bool continueOperation;

    mainMenu:

    while (choice != 'q')
    {
        continueOperation = true;
        cout << "Choose what you want to do in library ('b' for borrowing, 'r'
for returning, 'e' for enumerating, 'q' for quitting): ";
        cin >> choice;
        switch (choice)
        {
            case 'b':
            {
                do
                {
                    cout << "Write index number of book that you want to borrow,
or '100' for returning to menu: ";
                    cin >> borr;
                    if (borr >= 0 && borr < 10)
                    {
                        if (bookAvailability[borr])
                        {
                            bookAvailability[borr] = false;
                            cout << "You have borrowed \"" << books[borr] << "\"."
<< endl;
                        }
                        else
                        {

```

```

        cout << "The book \"" << books[borr] << "\" is already
borrowed." << endl;
    }
}
else if (borr == 100)
{
    continueOperation = false;
}
else
{
    cout << "Invalid book index!" << endl;
}
} while (continueOperation);
break;
}

case 'r':
{
    do
    {
        cout << "Write index number of book that you want to return or
'100' for returning to menu: ";
        cin >> ret;
        if (ret >= 0 && ret < 10)
        {
            if (!bookAvailability[ret])
            {
                bookAvailability[ret] = true;
                cout << "You have returned \"" << books[ret] << "\"."
<< endl;
            }
            else
            {
                cout << "The book \"" << books[ret] << "\" was not
borrowed." << endl;
            }
        }
        else if (ret == 100)
        {
            continueOperation = false;
        }
        else
        {
            cout << "Invalid book index!" << endl;
        }
        break;
    } while (continueOperation);
    break;
}

```



```

    }

    case 'e':
        for (int i = 0; i < 10; i++)
        {
            cout << "Book \"" << books[i] << "\" with index " << i << " is
";

            if (bookAvailability[i])
                cout << "available" << endl;
            else
                cout << "not available!" << endl;
        }

    case 'q':
        cout << "See you later!" << endl;
        break;

    default:
        cout << "Invalid choice! Please try again." << endl;
        goto mainMenu;
        break;
    }
}
}

```

Завдання №6

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-f01da8425fbc0753b1ea2edb71602a9eddad0ad4fc972f56034bfb3df4a5ce56

```

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

int main() {
    int n, winVasyl = 0, winKolia = 0, fullWinVasyl = 0, fullWinKolia = 0;
    string wins;

    cin >> n;
    cin >> wins;
    for (char i : wins) {
        (i == 'V') ? winVasyl += 1 : winKolia += 1;
        if ((winVasyl >= 11) && (winVasyl - winKolia >= 2)) {
            winVasyl = 0;

```

```

        winKolia = 0;
        fullWinVasyl += 1;
    }
    else if ((winKolia >= 11) && (winKolia - winVasyl >= 2)) {
        winVasyl = 0;
        winKolia = 0;
        fullWinKolia += 1;
    }
}
cout << fullWinKolia << ':' << fullWinVasyl << endl;
if ((winVasyl > 0) || (winKolia > 0))
    cout << winKolia << ':' << winVasyl;
}

```

5. Результати виконання завдань, тестування та фактично затрачений час:

Завдання №1

Виведемо рекурентну формулу $d = \frac{a_{n+1}}{a_n} = \frac{\frac{10^{n+1}}{(n+1)!}}{\frac{10^n}{n!}} = \frac{10}{n+1}$

Тепер щоб отримати наступний елемент потрібно помножити попередній елемент на d.

```

PS C:\Users\Lenovo\ai_programming_playground_2024\ai_12> & 'c:\Users\Lenovo\.vscode\extensions\ms-vscode.cpptools-1.22.10-win32-
bugLauncher.exe' '--stdin=Microsoft-MIEngine-In-dppla3lu.zkf' '--stdout=Microsoft-MIEngine-Out-cxnonwtz.evg' '--stderr=Microsoft-
-pid=Microsoft-MIEngine-Pid-5dmdpjl.lhk' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
The sum of the series with an accuracy of epsilon = 0.0001 is 22025.5 The value is reached at an = 3.38716e-05 when n = 33
PS C:\Users\Lenovo\ai_programming_playground_2024\ai_12>

```

Час затрачений на виконання завдання: 35 хв.

Завдання №2

Отримані результати можуть навести на висновок, що задане n, чи розмір епсілона не дуже змінюють результат. Проте самі степеневі ряди дуже сильно відрізняються від значення функції.

Особливості виконання роботи було те що, бібліотека smath не надає функції $\operatorname{arccot}(x)$, тому я її замінив рівною функцією $\arctan(1/x)$.

```

PS C:\Users\Lenovo\ai_programming_playground_2024\ai_12>
bugLauncher.exe' '--stdin=Microsoft-MIEngine-In-zjevp3oa.|
-pid=Microsoft-MIEngine-Pid-hxtwca3a.ztv' '--dbgExe=C:\ms
X = 0.1 SN = 0.100002 SE = 0.100002 Y = 0.785732
X = 0.17 SN = 0.170028 SE = 0.170028 Y = 0.787036
X = 0.24 SN = 0.24016 SE = 0.24016 Y = 0.790013
X = 0.31 SN = 0.310576 SE = 0.310576 Y = 0.795368
X = 0.38 SN = 0.381603 SE = 0.381603 Y = 0.803854
X = 0.45 SN = 0.453777 SE = 0.453775 Y = 0.816321
X = 0.52 SN = 0.527929 SE = 0.527929 Y = 0.833808
X = 0.59 SN = 0.605342 SE = 0.605342 Y = 0.857714
X = 0.66 SN = 0.688034 SE = 0.688084 Y = 0.890118
X = 0.73 SN = 0.779289 SE = 0.779632 Y = 0.934473
X = 0.8 SN = 0.884678 SE = 0.886646 Y = 0.997334
PS C:\Users\Lenovo\ai_programming_playground_2024\ai_12>

```

Час затрачений на виконання завдання: 40 хв.

Завдання №3

Наведені приклади з методички використовують не правильні методи для c++, тому довелося використати `va_list` для задання змінної кількості аргументів в функції та шаблони для використання різних типів змінних в функції.

```

PS C:\Users\Lenovo\ai_programming_playground_2024\ai_12>
bugLauncher.exe' '--stdin=Microsoft-MIEngine-In-livcx4fs.
-pid=Microsoft-MIEngine-Pid-gbqkd333.qis' '--dbgExe=C:\ms
-4
-69.23
-45
PS C:\Users\Lenovo\ai_programming_playground_2024\ai_12>

```

Завдання №4

Для додавання десяткових чисел створив перевантажені функцію, що сприймає `double` і `float` типи змінних. Звичайні дробі вирішив задати, як пари `int` або `long long`, та створив відповідну перевантажену функцію.

```

PS C:\Users\Lenovo\ai_programming_playground_2024\ai_12> & '
bugLauncher.exe' '--stdin=Microsoft-MIEngine-In-4izo2g5c.hdm'
-pid=Microsoft-MIEngine-Pid-lmfws13.5az' '--dbgExe=C:\msys64
2279.78
2279.78
91/12
8303420137191222087/1316947263937829686
PS C:\Users\Lenovo\ai_programming_playground_2024\ai_12>

```

Завдання №5

- **Ініціалізація:**

- Програма починається з ініціалізації масивів `books` та `bookAvailability`, де зберігаються назви книг та їх доступність відповідно. Масив `books` містить назви 10 книг, а масив `bookAvailability` має булеві значення, які вказують, чи доступна відповідна книга (значення `true` означає, що книга доступна).

- **Меню:**

- Користувачу пропонується вибрати одну з опцій (створено за допомогою `while` та `switch case`):
 - 'b' - взяти книгу.
 - 'r' - повернути книгу.
 - 'e' - переглянути доступність книг.
 - 'q' - завершити роботу програми.
- Після вибору користувачем опції, програма виконує відповідну дію.

- **Позичання книги (створено за допомогою `do while`):**

- Якщо обрано 'b', користувачу пропонується ввести індекс книги, яку він хоче позичити.
- Якщо книга доступна, то її статус змінюється на `false`, і користувач отримує повідомлення про успішне позичання.
- Якщо книга вже позичена, програма повідомляє про це.
- Для повернення до меню користувач може ввести 100.

- **Повернення книги (створено за допомогою `do while`):**

- Якщо обрано 'r', користувачу пропонується ввести індекс книги, яку він хоче повернути.
- Якщо книга позичена (статус `false`), вона змінюється на `true`, і програма виводить повідомлення про успішне повернення.
- Якщо книга не була позичена, програма повідомляє про це.
- Для повернення до меню користувач може ввести 100.

- **Перегляд наявності книг (створено за допомогою `for`)**

- Опція 'e' дозволяє користувачу переглянути поточний статус кожної книги, вказуючи, чи доступна книга.

- **Вихід із програми (створено за допомогою `goto`):**

- Вибір 'q' завершує роботу програми та виводить повідомлення "See you later!".

Завдання №6

При кожному виграші очка ('V' для Василя і 'К' для Колі) програма перевіряє, чи є переможець партії. Якщо так, очки обнуляються, і збільшується лічильник виграних партій. Виводяться результати виграних партій і, якщо є незакінчена партія, поточні очки.

5 hours ago 0031 - Коля, Вася і Теніс C++ 23 Accepted 0.005 1.180 1859483

6. Кооперація з командою:

- Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Тrello

The top part of the image shows a Zoom meeting window. The main display is a presentation of a C++ program titled "Epic 3". The program calculates the future value of an investment based on principal amount, annual percentage rate, and time. The code is as follows:

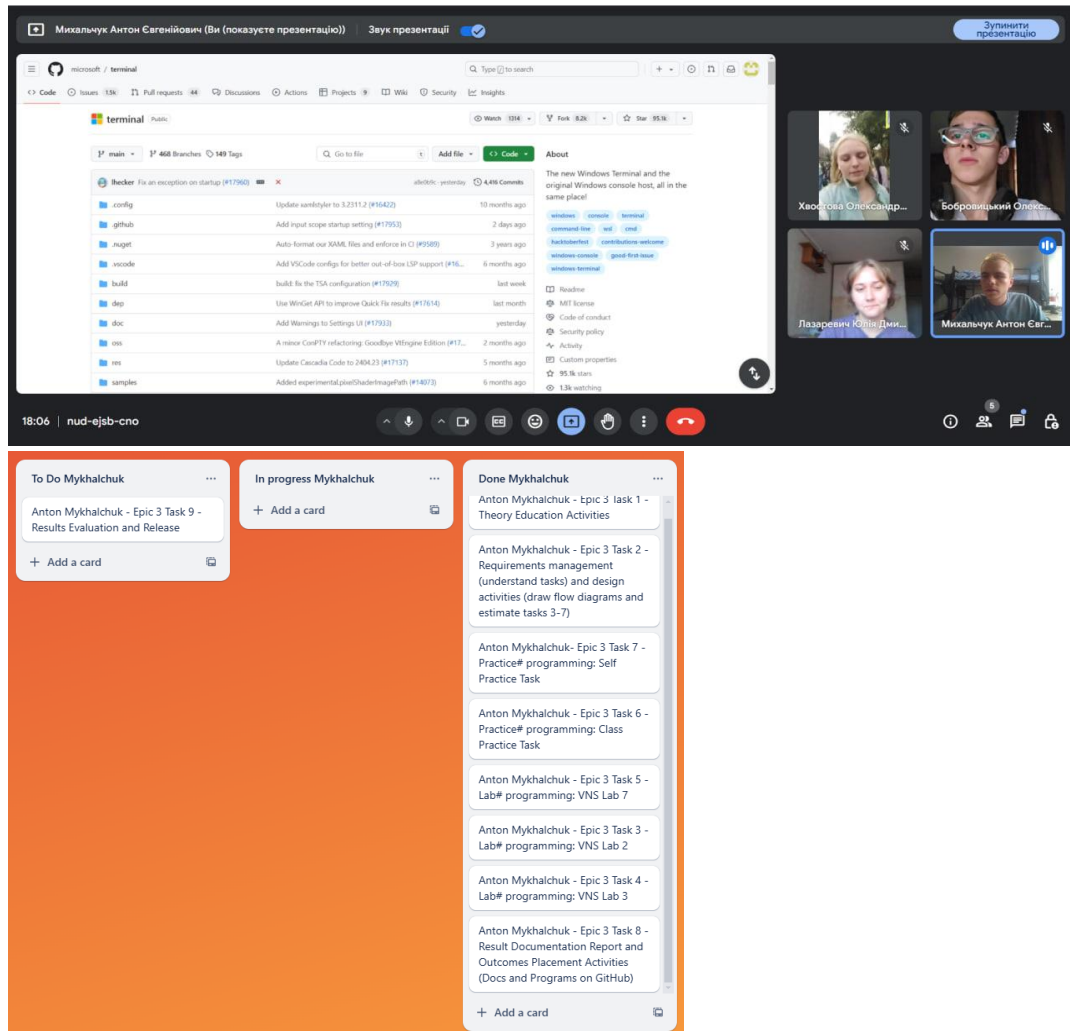
```

1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 int main() {
6     float a, p, r;
7     int n, t;
8     printf("Enter principal amount of investment: ");
9     scanf("%f", &p);
10
11     printf("Enter annual percentage rate: ");
12     scanf("%f", &r);
13
14     printf("Enter number of interest charges per year: ");
15     scanf("%d", &n);
16
17     printf("Enter time for which the money is invested, in years: ");
18     scanf("%d", &t);
19
20     // ... (rest of the code)
21
22     return 0;
23 }
```

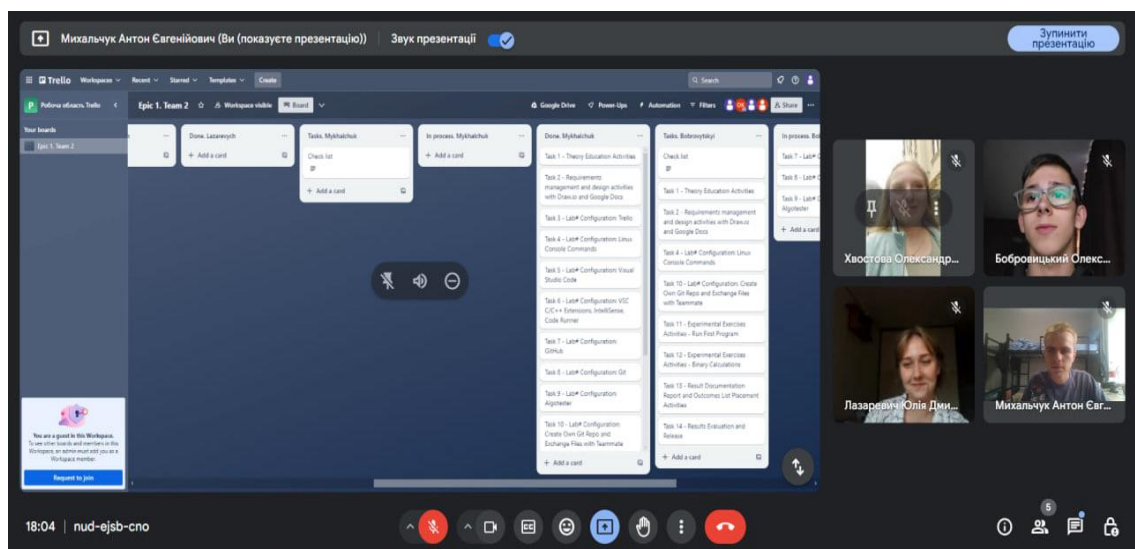
The bottom part of the image shows a Trello board with three columns: "To Do Mykhalchuk", "In progress Mykhalchuk", and "Done Mykhalchuk".

- To Do Mykhalchuk:**
 - Anton Mykhalchuk - Epic 3 Task 9 - Results Evaluation and Release
- In progress Mykhalchuk:**
 - + Add a card
- Done Mykhalchuk:**
 - Anton Mykhalchuk - Epic 3 Task 1 - Theory Education Activities
 - Anton Mykhalchuk - Epic 3 Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)
 - Anton Mykhalchuk - Epic 3 Task 7 - Practice# programming: Self Practice Task
 - Anton Mykhalchuk - Epic 3 Task 6 - Practice# programming: Class Practice Task
 - Anton Mykhalchuk - Epic 3 Task 5 - Lab# programming: VNS Lab 7
 - Anton Mykhalchuk - Epic 3 Task 3 - Lab# programming: VNS Lab 2
 - Anton Mykhalchuk - Epic 3 Task 4 - Lab# programming: VNS Lab 3
 - Anton Mykhalchuk - Epic 3 Task 8 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

- Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло



- Скрін з 3-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло (опційно)



To Do Mykhalchuk ...

- Anton Mykhalchuk - Epic 3 Task 9 - Results Evaluation and Release

+ Add a card

In progress Mykhalchuk ...

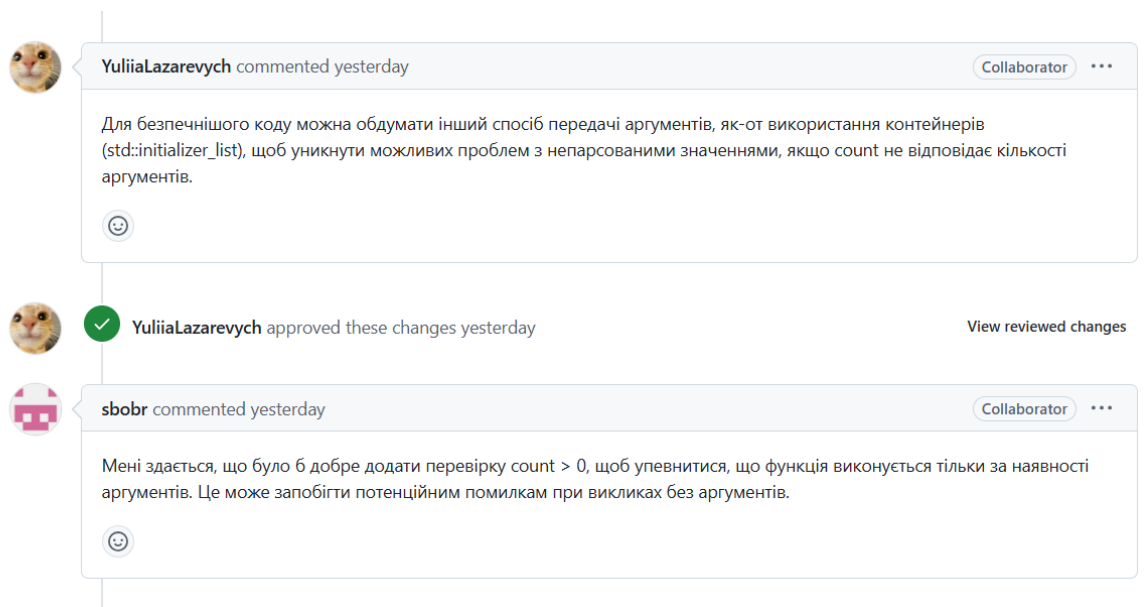
+ Add a card

Done Mykhalchuk ...

- Anton Mykhalchuk - Epic 3 Task 1 - Theory Education Activities
- Anton Mykhalchuk - Epic 3 Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)
- Anton Mykhalchuk- Epic 3 Task 7 - Practice# programming: Self Practice Task
- Anton Mykhalchuk - Epic 3 Task 6 - Practice# programming: Class Practice Task
- Anton Mykhalchuk - Epic 3 Task 5 - Lab# programming: VNS Lab 7
- Anton Mykhalchuk - Epic 3 Task 3 - Lab# programming: VNS Lab 2
- Anton Mykhalchuk - Epic 3 Task 4 - Lab# programming: VNS Lab 3
- Anton Mykhalchuk - Epic 3 Task 8 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

+ Add a card

- Скрін з 2-му коментарями від учасників команди на пул реквесті з Рев'ю Роботи



Висновки:

У ході роботи було розглянуто та опрацьовано основні концепції програмування в мові C++, пов'язані з циклами та функціями. Особливу увагу приділено вкладеним циклом, умовам завершення їх виконання, а також реалізації різноманітних завдань за допомогою циклічних структур.

Засвоєно принципи роботи з функціями, включаючи базові виклики та перевантаження для підтримки аргументів різного типу. Досліджено використання функцій зі змінною кількістю параметрів, рекурсивні виклики та вбудовані функції для оптимізації коду. Виконані завдання дозволили поглибити розуміння застосування теоретичних знань на практиці, включаючи реалізацію математичних розрахунків через ряди, обчислення з точністю до ϵ , а також створення програм із використанням структурованих та перевантажених функцій для специфічних сценаріїв.

Загалом, робота дала змогу оволодіти важливими аспектами циклів і функцій, що є основою для створення ефективних програм на C++.