Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

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

до:

ВНС Лабораторної Роботи № 2

ВНС Лабораторної Роботи № 3

ВНС Лабораторної Роботи № 7

Практичних Робіт № 3

**Виконав:**

Студент групи ШІ-11

Ковалець Владислав Миколайович

# **Тема роботи:**

" Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія"

# **Мета роботи:**

Створення програми, використовуючи вкладені цикли, функції, перевантаження функцій та рекурсію

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли. Вкладені Цикли
* Тема №2: Функції. Перевантаження функцій
* Тема №3: Рекурсія

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли

Джерела Інформації:

* + - Відео:

https://www.youtube.com/watch?v=zBtcqNdiRf4

* + Що опрацьовано:
    - Синтаксис циклів в с++.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 11.11.2023
  + Звершення опрацювання теми: 11.11.2023
* Тема №2: Функції. Перевантаження функцій
  + Джерела Інформації:
    - Відео:

https://www.youtube.com/watch?v=3KJfisev6SI

* + Що опрацьовано:
    - Розглянуто функції та перевантаження функції
  + Статус: Ознайомлений
  + Початок опрацювання теми: 12.10.2023
  + Звершення опрацювання теми: 12.10.2023
* Тема №3: Рекурсія
  + Джерела Інформації:
    - Відео

<https://www.youtube.com/watch?v=V7q9w_s0nns>

* + Що опрацьовано:
    - Ознайомлений з рекурсією функції та її застосуванням
  + Статус: Ознайомлений
  + Початок опрацювання теми: 12.10.2023
  + Звершення опрацювання теми: 12.10.2023

# **Виконання роботи:**

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

Завдання № VNS lab2

* 2 варіант

Знайти суму цілих додатніх парних чисел, менших 100.

Важливі деталі для врахування в імплементації програми:

Використовуючи оператор циклу, знайти суму елементів, зазначених у

конкретному варіанті. Результат надрукувати, надавши відповідний заголовком.

Завдання № VNS lab3

* 2 варіант

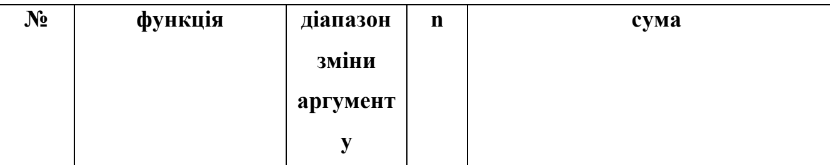
Для х, що змінюється від a до b з кроком (b-a)/k, де (k=10), обчислити

функцію f(x), використовуючи її розклад в степеневий ряд у двох випадках:

а) для заданого n;

б) для заданої точності ε (ε=0.0001).

Для порівняння знайти точне значення функції.



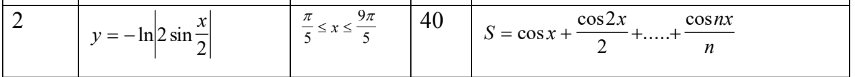


Рисунок 1 Умови задачі

Завдання № VNS lab7

* 2 варіант task1

Написати функцію mult зі змінною кількістю параметрів, що знаходить

добуток чисел типу float. Написати викликаючу функцію main, що

звертається до функції mult не менше трьох разів з кількістю параметрів 3, 7, 11.

* 2 варіант task2

Написати перевантажені функції й основну програму, що їх викликає.

а) для додавання дійсних чисел;

б) для додавання комплексних чисел.

Завдання № practice №3

* Загальний варіант

Epic 3 Practice - Цикли

Task Deskription

Менеджмент бібліотеки

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.

Вимоги

⁃ while: продовжувати працювати, доки користувач не вирішить вийти.

⁃ do while: Після кожної операції (позичити, повернути, перерахувати) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.

⁃ for: список усіх книг за допомогою циклу for.

⁃ для кожної: перевірити наявність кожної книги.

⁃ goto: якщо користувач вводить неправильний вибір, використовуйте goto, щоб перенаправити його до головного меню.

* **Важливі деталі для врахування в імплементації програми:**

Використовувати дані цикли.

Завдання № selfpractice

* Варіант –

Депутатські гроші

Обмеження: 2 сек., 256 МіБ

Часто-густо громадяни намагаються з’ясувати, наскільки багатими є депутати. Дехто вірить, що матеріальні статки окремих депутатів є необмеженими. Тож уявіть собі депутата, у якого є необмежена кількість купюр усіх номіналів (1, 2, 5, 10, 20, 50, 100, 200 та 500 гривень). Він хоче придбати подарунок для своєї тещі, що коштує n гривень. Незважаючи на те, що наш герой-олігарх є нескінченно багатим, він також є нескінченно скупим та педантичним. Саме тому він хоче оплатити покупку готівкою без решти. Яка мінімальна кількість купюр йому для цього знадобиться?

Вхідні дані

У єдиному рядку задано одне натуральне число n— вартість подарунку.

Вихідні дані

У єдиному рядку виведіть одне ціле число — мінімальну кількість купюр, що необхідна для покупки подарунка.

Обмеження

1≤n≤10^9

## **2. Дизайн та планована оцінка часу виконання завдань:**

Програма № VNS lab2

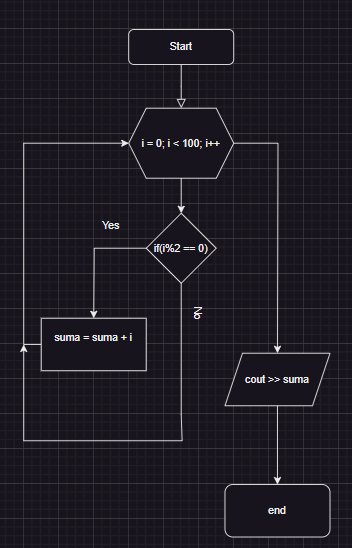
* Блок-схема:
* 

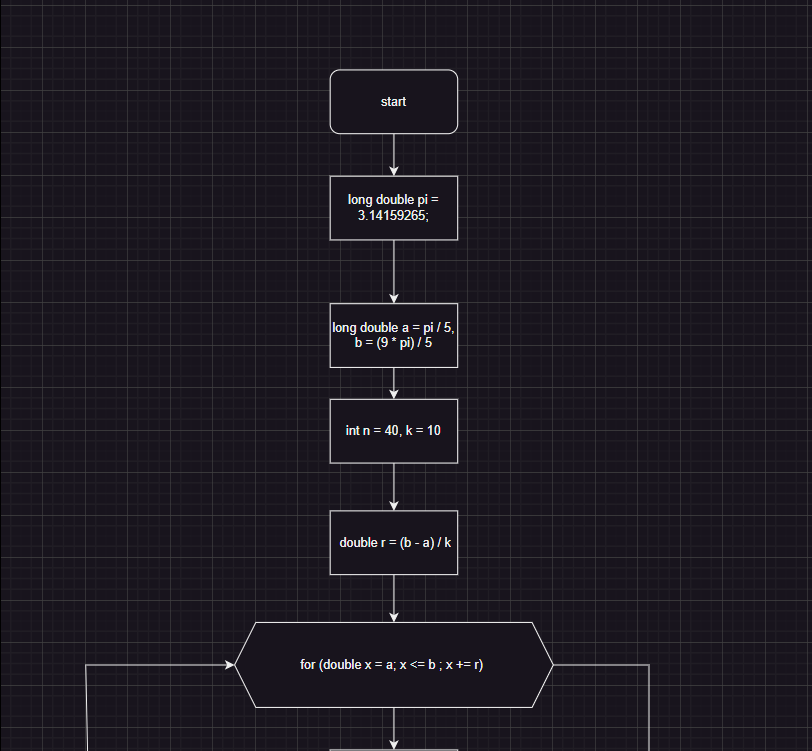
Рисунок 2 Блок-схема до vns lab2

* Планований час на реалізацію

10 хвилин

Програма № vns lab3

* Блок-схема a)



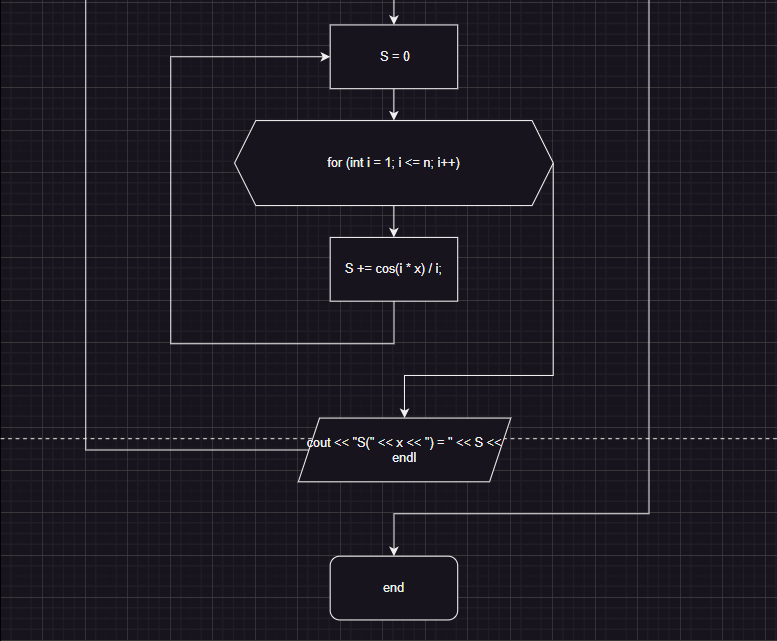
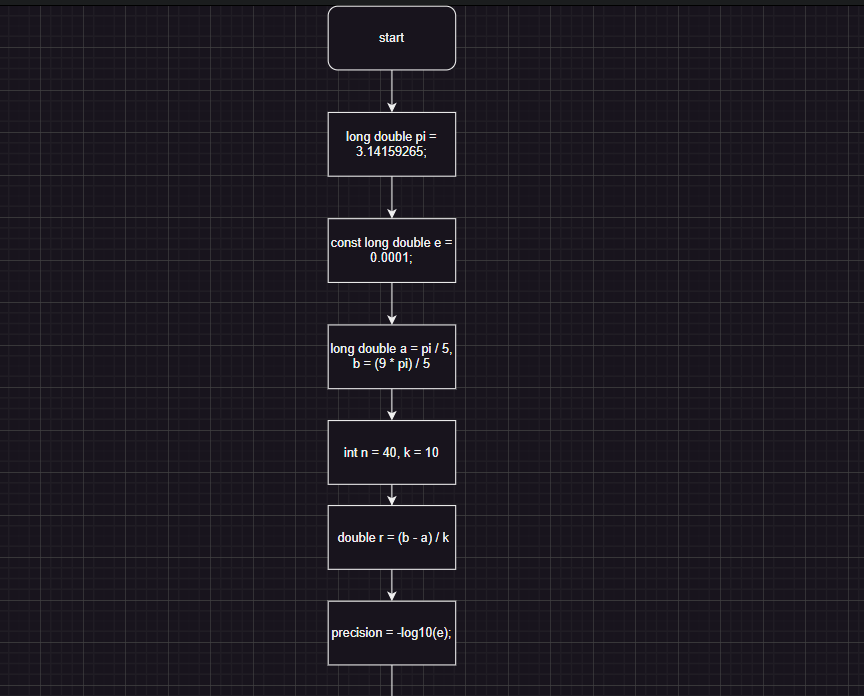


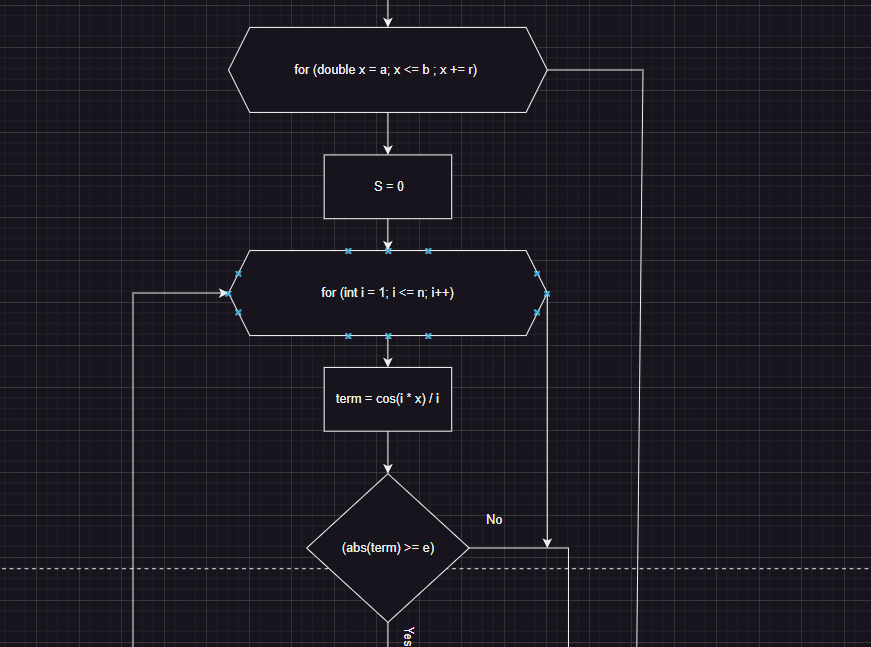
Рисунок 3 Блок-схема до vns lab3 task1

* Планований час на реалізацію

30 хвилин

* Блок-схема б)





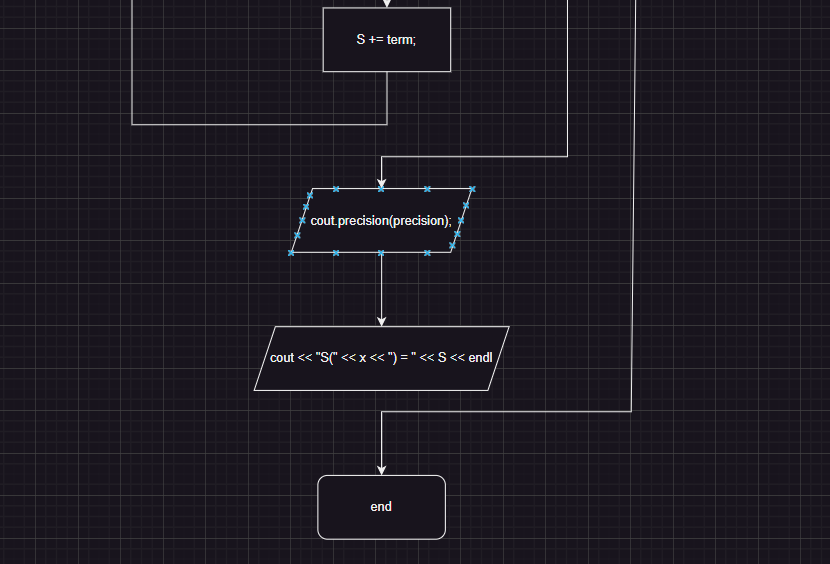


Рисунок 4 Блок-схема до vns lab3 task2

Програма № vns lab7 task1

* Блок-схема

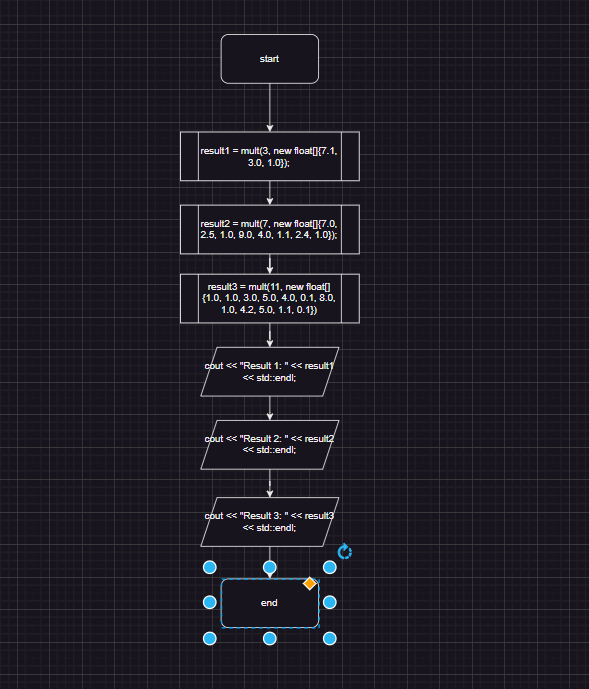


Рисунок 5 Блок-схема до vns lab7 task1

* Планований час на реалізацію

30 хвилин

Програма № vns lab7 task2

* Блок-схема

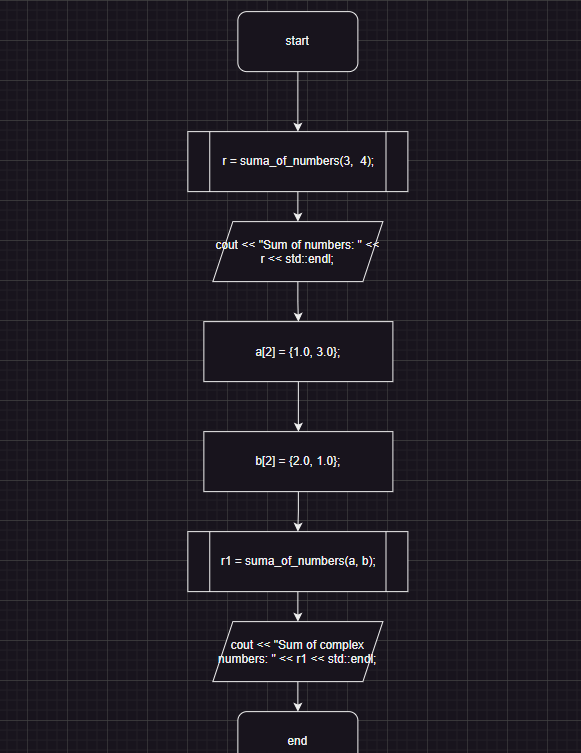


Рисунок 6 Блок-схема до vns lab7 task2

* Планований час на реалізацію

30 хвилин

Програма № practice3

* Блок-схема

**ai\_programming\_playground\ai\_11\vladyslav\_kovalets\epic3\** **практична3.draw.io**

* Планований час на реалізацію

30 хвилин

Програма № selfpractice

* Блок-схема

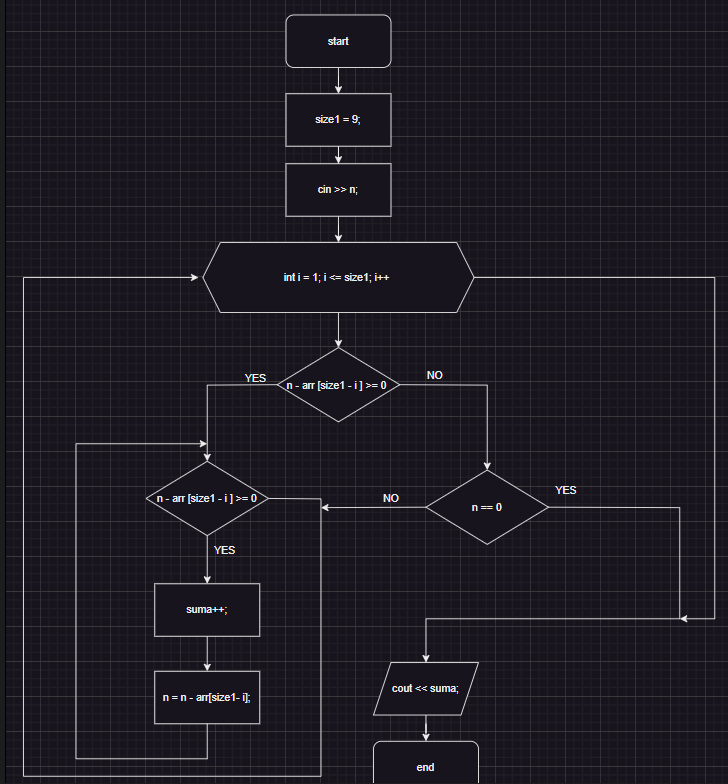


Рисунок 7 Блок-схема до selfpractice

* Планований час на реалізацію

20 хвилин

* Важливі деталі для врахування в імплементації

## **3. Конфігурація середовища до виконання завдань:**

Завдання № - Деталі по конфігурації середовища + скріншоти з підписами до скріншотів.

**Операційна система**: Windows

**Інсталяція інтерпретаторів або компіляторів: С++**

**Інсталяція залежностей: -**

**Інтегроване середовище розробки (IDE)**: Visual Studio Code

**Зовнішні ресурси**:



Рисунок 8 Завантажені бібліотеки або плагіни

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

**Завдання № vns lab2**

Знайти суму цілих додатніх парних чисел, менших 100.

#include <iostream>

int main()

{

    int suma;

    for(int i = 0; i < 100; i++)

    {

        if(i % 2 == 0)

        suma = suma + i;

    }

    std::cout << "Suma of the first 100 even numbers: " << suma;

    return 0;

}

**Завдання № practice 3**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main()

{

    vector<string> books = {"Harry\_Potter", "Hobbit", "Pride\_and\_Prejudice", "1984"};

    vector<bool> availability(books.size(), true);

    string userstring, takebooks, takemenu, mainmenu, bookscheks;

    string spusok("spusok"), take("take"), returnbook("return"), exit\_program("exit");

    bool exitprogram = true;

    label1:

    while (exitprogram)

    {

        cout << "Enter the action (Example: spusok, take, return, exit): ";

        cin >> userstring;

        if (userstring == spusok)

        {

            cout << "List of available books:" << endl;

            for (int i = 0; i < books.size(); i++)

            {

                if (availability[i])

                    cout << books[i] << " (Available)" << endl;

                else

                    cout << books[i] << " (Not Available)" << endl;

            }

            cout << "Do you want to return to the main menu (yes / no): ";

            cin >> mainmenu;

            if (mainmenu == "no")

            {

                exitprogram = false;

            }

        }

        else if (userstring == take)

        {

            cout << "Enter the name of the book you want to take( enter with \_ ): ";

            cin >> takebooks;

            bool bookTaken = false;

            for (int i = 0; i < books.size(); i++)

            {

                if (books[i] == takebooks && availability[i])

                {

                    cout << "You have taken the book: " << books[i] << endl;

                    availability[i] = false;

                    bookTaken = true;

                    break;

                }

            }

            if (bookTaken == false) //  перевірка чи книгу вдалося взяти

            {

                cout << "Book not found or already taken." << endl;

            }

            cout << "Do you want to return to the main menu (yes / no): ";

            cin >> mainmenu;

            if (mainmenu == "no")

            {

                exitprogram = false;

            }

            else

            {

                userstring = "";

            }

        }

        else if (userstring == returnbook)

        {

            cout << "Enter the name of the books you want to return (Enter with \_): ";

            cin >> bookscheks;

            bool bookTaken = true;

            for(int i = 0; i < books.size(); i++)

            {

                if (books[i] == bookscheks && availability[i] == false)

                {

                    cout << "You have return the book: " << books[i] << endl;

                    availability[i] = true;

                    bookTaken = false;

                    break;

                }

            }

            if (bookTaken == true)

            {

                cout << "Book not found or not taken." << endl;

            }

            cout << "Do you want to return to the main menu (yes / no): ";

            cin >> mainmenu;

            if (mainmenu == "no")

            {

                exitprogram = false;

            }

            else

            {

                userstring = "";

            }

        }

        else if (userstring == exit\_program)

        {

            exitprogram = false;

        }

        else

        {

            cout << "Undefined command" << endl;

            goto label1;

        }

    }

    return 0;

}

**Завдання № vns lab3 task1**

1. Розв’язати задачу

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

    long double S = 0;

    const long double pi = 3.14159265358979323846;

    const long double a = pi / 5, b = (9 \* pi) / 5;

    const int n = 40, k = 10; // при заданому n

    long double r = (b - a) / k;

    for (double x = a; x <= b ; x += r)

    {

        S = 0;

        for (int i = 1; i <= n; i++)

        {

            S += cos(i \* x) / i;

        }

        cout << "S(" << x << ") = " << S << endl;

    }

    return 0;

}

**Завдання № vns lab3 task2**

1. Розв’язати задачу

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

    long double S = 0;

    const long double pi = 3.14159265358979323846;

    const long double e = 0.0001;

    long double a = pi / 5, b = (9 \* pi) / 5;

    int n = 4000, k = 10;

    long double r = (b - a) / k;

    // Визначення кількості значень після коми

    int precision = -log10(e);

    for (long double x = a; x <= b + r / 2; x += r)

    {

        S = 0;

        for (int i = 1; i <= n; i++)

        {

            long double term = cos(i \* x) / i;

            if (abs(term) >= e)

            {

                S += term;

            } else

            {

                break;

            }

        }

        // Виведення з використанням точності е

        cout.precision(precision);

        cout << "S(" << x << ") = " << S << endl;

    }

    return 0;

}

**Завдання № vns lab7 task1**

1. Розв’язати задачу

#include <iostream>

double mult(int count, float args[])

{

    float dobutok = 1;

    for (int i = 0; i < count; ++i)

    {

        dobutok = dobutok \* args[i];

    }

    return dobutok;

}

int main()

{

    float result1 = mult(3, new float[]{7.1, 3.0, 1.0});

    float result2 = mult(7, new float[]{7.0, 2.5, 1.0, 9.0, 4.0, 1.1, 2.4, 1.0});

    float result3 = mult(11, new float[]{1.0, 1.0, 3.0, 5.0, 4.0, 0.1, 8.0, 1.0, 4.2, 5.0, 1.1, 0.1});

    std::cout << "Result 1: " << result1 << std::endl;

    std::cout << "Result 2: " << result2 << std::endl;

    std::cout << "Result 3: " << result3 << std::endl;

    return 0;

}

**Завдання № vns lab7 task2**

1. Розв’язати задачу

#include <iostream>

#include <complex>

double suma\_of\_numbers(int a, int b)

{

    return a + b;

}

std::complex<double> suma\_of\_numbers(double a[2], double b[2])

{

    std::complex<double> z1(a[0], b[0]); // 1 + 2i

    std::complex<double> z2(a[1], b[1]); // 3 + 1i

    return z1 + z2;

}

int main()

{

    double r = suma\_of\_numbers(3, 4);

    std::cout << "Sum of numbers: " << r << std::endl;

    double a[2] = {1.0, 3.0};

    double b[2] = {2.0, 1.0};

    std::complex<double> r1 = suma\_of\_numbers(a, b);

    std::cout << "Sum of complex numbers: " << r1 << std::endl;

    return 0;

}

**Завдання № selfpractice**

#include <iostream>

using namespace std;

int main()

{

    int arr[]{1, 2, 5, 10, 20, 50, 100, 200, 500};

    int size1 = 9;

    long long n, suma = 0;

    cin >> n;

    for(int i = 1; i <= size1; i++ )

    {

        if (n - arr [size1 - i ] >= 0 )

        {

            while(n - arr [size1 - i ] >= 0)

            {

            suma++;

            n = n - arr[size1- i];

            }

        }

        else if (n == 0)

        {

            break;

        }

    }

    cout << suma;

}

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

Завдання № vns lab2

Програма рахує суму парних чисел <100

Час затрачений на виконання завдання

10 хвилин

Завдання № practice 3

Програма представляє собою текстовий інтерфейс для управління бібліотекою книг. Користувач може переглядати список доступних книг (spusok), взяти книгу (take), повернути книгу (return), або вийти з програми (exit). Для цього використовуються вектори для зберігання імен книг та їх доступності.

Час затрачений на виконання завдання

20 хвилин

Завдання № vns lab3 task1

Даний код реалізує обчислення суми S(x) для кожного значення x у визначеному інтервалі [a,b], де a=π\5 та b=9π\5 Кожна ітерація циклу по x обчислює відповідну суму, яка складається з n елементів, де n=40. Всі значення S(x) виводяться на екран

Програма використовує константи для збереження значень π, a, та b, а також використовує r для розрахунку кроку між значеннями x у заданому інтервалі.

Час затрачений на виконання завдання

30 хвилин

Завдання № vns lab3 task2

Програма обчислює суму S(x) для кожного значення x у визначеному інтервалі [a,b], але з використанням додаткової умови зупинки внутрішнього циклу, яка враховує точність e=0.0001. Також, виведення значень суми S(x) тепер відбувається з використанням точності e.

Час затрачений на виконання завдання

40 хвилин

Завдання № vns lab7 task1

В даній програмі визначена функція mult, яка обчислює добуток переданих їй чисел з використанням динамічного виділення пам'яті для передачі масиву аргументів.

Час затрачений на виконання завдання

30 хвилин

Завдання № vns lab7 task2

Програма містить дві функції suma\_of\_numbers та suma\_of\_i, які виконують додавання двох цілих чисел та двох комплексних чисел відповідно. В функції suma\_of\_numbers передаються два цілих числа, їхню суму обчислюється та повертається.

У функції suma\_of\_i створюються два комплексних числа z1 та z2, їхню суму обчислюється та повертається.

В main викликаються обидві функції, і результати виводяться на екран.

Програма демонструє використання стандартної бібліотеки С++ для роботи з комплексними числами та простими арифметичними операціями.

Час затрачений на виконання завдання

30 хвилин

Завдання № selfpractice

Програма реалізує жадий алгоритм для знаходження мінімальної кількості купюр, які необхідно видати, щоб зібрати задану суму n з використанням заданого масиву купюр.



Рисунок 9 Підтвердження перевірки коду на Algotester

Час затрачений на виконання завдання

30 хвилин

# **Висновки:**

Під час виконання роботи з темою "Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія" було проведено дослідження та робота з різними аспектами програмування в мові C++. Зокрема, було опрацьовано три основні теми:

1. \*\*Цикли та вкладені цикли:\*\*

- Ознайомлено з синтаксисом циклів в мові C++.

- Розглянуто відео-матеріали, що надають базові знання щодо циклів та їх використання.

2. \*\*Функції та перевантаження функцій:\*\*

- Детально вивчено концепції функцій та їх перевантаження.

- Розглянуто відео-матеріали, які демонструють використання функцій та перевантаження функцій у C++.

3. \*\*Рекурсія:\*\*

- Ознайомлено з концепцією рекурсії в програмуванні.

- Вивчено відео-матеріали, які демонструють застосування рекурсії в функціях.

Завдання, які було виконано під час роботи:

- Створено програму для знаходження суми цілих додатніх парних чисел, менших 100, використовуючи цикли.

- Розроблено програму для обчислення функції \(f(x)\), використовуючи її розклад в степеневий ряд для заданого \(n\) та з точністю \(ε\).

- Написано функцію mult зі змінною кількістю параметрів та створено викликаючу функцію main, яка звертається до функції mult не менше трьох разів з різною кількістю параметрів.

Крім того, було вирішено практичне завдання з керування бібліотекою, використовуючи цикли для перерахування книг та перевірки їх доступності.

В процесі виконання роботи було використано різні джерела інформації, включаючи відео-уроки та інші онлайн-ресурси.

У результаті роботи було набуто практичних навичок роботи з циклами, функціями, перевантаженням функцій та рекурсією в мові програмування C++.