Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт №3**

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

***з розділу***: «Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія.»

***Виконала:***

студентка групи ШІ-13

Новосад Олена Петрівна

# **Тема роботи:**

Ознайомлення з різними видами циклів, концептом вкладених циклів, поняттям функції, перевантаженням функції і практичністю застосування цього методу, а також рекурсією.

# **Мета роботи:**

Детальніше ознайомитися із поняттям функції, її перевантаженням, а також доцільністю застосування цього методу. Дослідити інформацію щодо видів циклів, а також застосування вкладених циклів при створенні програмного коду. Дізнатися більше про рекурсію. Після опрацювання інформації застосувати здобуті знання на практиці для кращого засвоєння.

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли. Вкладені цикли.
* Тема №2: Функція. Перевантаження функції.
* Тема №3: Рекурсія.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли. Вкладені цикли.
  + Джерела Інформації
    - [https://acode.com.ua/urok-69-operator-goto/](https://acode.com.ua/urok-69-operator-goto/%20)
    - <https://acode.com.ua/urok-70-tsykl-while/>
    - <https://acode.com.ua/urok-71-tsykl-do-while/>
    - <https://acode.com.ua/urok-72-tsykl-for/>
    - <https://acode.com.ua/urok-73-operatory-break-i-continue/>
    - <https://acode.com.ua/urok-95-tsykl-foreach/>
    - Конспект лекцій з дисципліни у VNS
  + Що опрацьовано:
    - Опрацьовано інформацію про види циклів for, while, do while, for each, а також про оператори break, сontinue, goto. Досліджено та застосовано на практиці поняття вкладених циклів.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.11.23
  + Звершення опрацювання теми: 28.11.23
* Тема №2: Функція. Перевантаження функції.
  + Джерела Інформації:
    - <https://acode.com.ua/urok-15-funktsiyi-i-operator-return/>
    - <https://acode.com.ua/urok-16-parametry-i-argumenty-funktsij/>
    - <https://acode.com.ua/urok-102-parametry-i-argumenty-funktsij/>
    - <https://acode.com.ua/urok-103-peredacha-po-znachennyu/>
    - <https://acode.com.ua/urok-104-peredacha-po-posylannyu/>
    - <https://acode.com.ua/urok-105-peredacha-po-adresu/>
    - <https://acode.com.ua/urok-108-perevantazhennya-funktsij/>
    - Лекційний матеріал до теми №7 у VNS
  + Що опрацьовано:
    - Детальніше ознайомилася з поняттям функції, її аргументів та параметрів, для чого її можна застосовувати, та як передавати значення аргументів у функцію. Також було опрацьовано матеріал по перевантаженню функції.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 10.11.2023
  + Звершення опрацювання теми: 29.11.2023
* Тема №3: Рекурсія.
  + Джерела Інформації:
    - <https://acode.com.ua/urok-113-rekursiya-i-chysla-fibonachchi/>
  + Що опрацьовано:
    - Опрацювала, що таке рекурсія та інформацію про рекурсивну функцію у С++ і її правильне застосовування.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 26.11.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 2

* 16 варіант
* Використовуючи оператор циклу, знайти суму елементів, зазначених у

конкретному варіанті. Результат надрукувати, надавши відповідний

заголовком.

* 16) Знайти суму ряду з точністю ε=0.0001, загальний член якого



* Ми зможемо дізнатися, що точність досягнута, коли |an|<E;
* Використовуємо рекурентну формулу для обчислень.

Завдання №2 VNS Lab 3

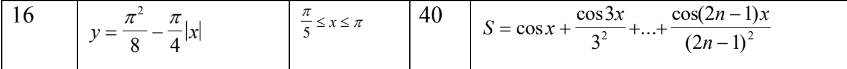
* 16 варіант
* Для х, що змінюється від a до b з кроком (b-a)/k, де (k=10), обчислити

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.



* Застосовуємо вкладені цикли, точність буде вважатися досягнутою, коли |se|<E (se – один з доданків ряду). Зовнішній цикл організує зміну параметра х.

Завдання №3 VNS Lab7 Task 1

* 16 варіант
* Розв’язати зазначене у варіанті завдання, використовуючи функції зі змінною

кількістю параметрів.

Написати функцію days зі змінною кількістю параметрів, яка знаходить

кількість днів, що пройшли між двома датами (параметрами функції є дати у

форматі «дд.мм.рр». Написати викликаючу функцію main, що звертається до

функції days не менше трьох разів з кількістю параметрів 3, 5, 8.

* Важливі деталі для врахування в імплементації програми: необхідно перевести дати в секунди, потім шукати різницю в секундах і отриману різницю поділити на кількість секунд в добі.

Завдання №4 VNS Lab7 Task 2

* 16 варіант
* Написати перевантажені функції й основну програму, що їх викликає.

а) для перетворення годин і хвилин у хвилини;

б)для перетворення хвилин у години та хвилини.

**Завдання №5 Class Practice Work «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.

**Мета Задачі**

Навчитися користуватися операторами циклів та функцією переходу на мітку:

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

**Важливі деталі для врахування в імплементації програми**

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 Task «Депутатські гроші»**

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?

Деталі імплементації:

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 №1 VNS Lab 2

* Планований час на реалізацію: 2 дні;

Програма №2 VNS Lab 3

* Планований час на реалізацію:2 дні;

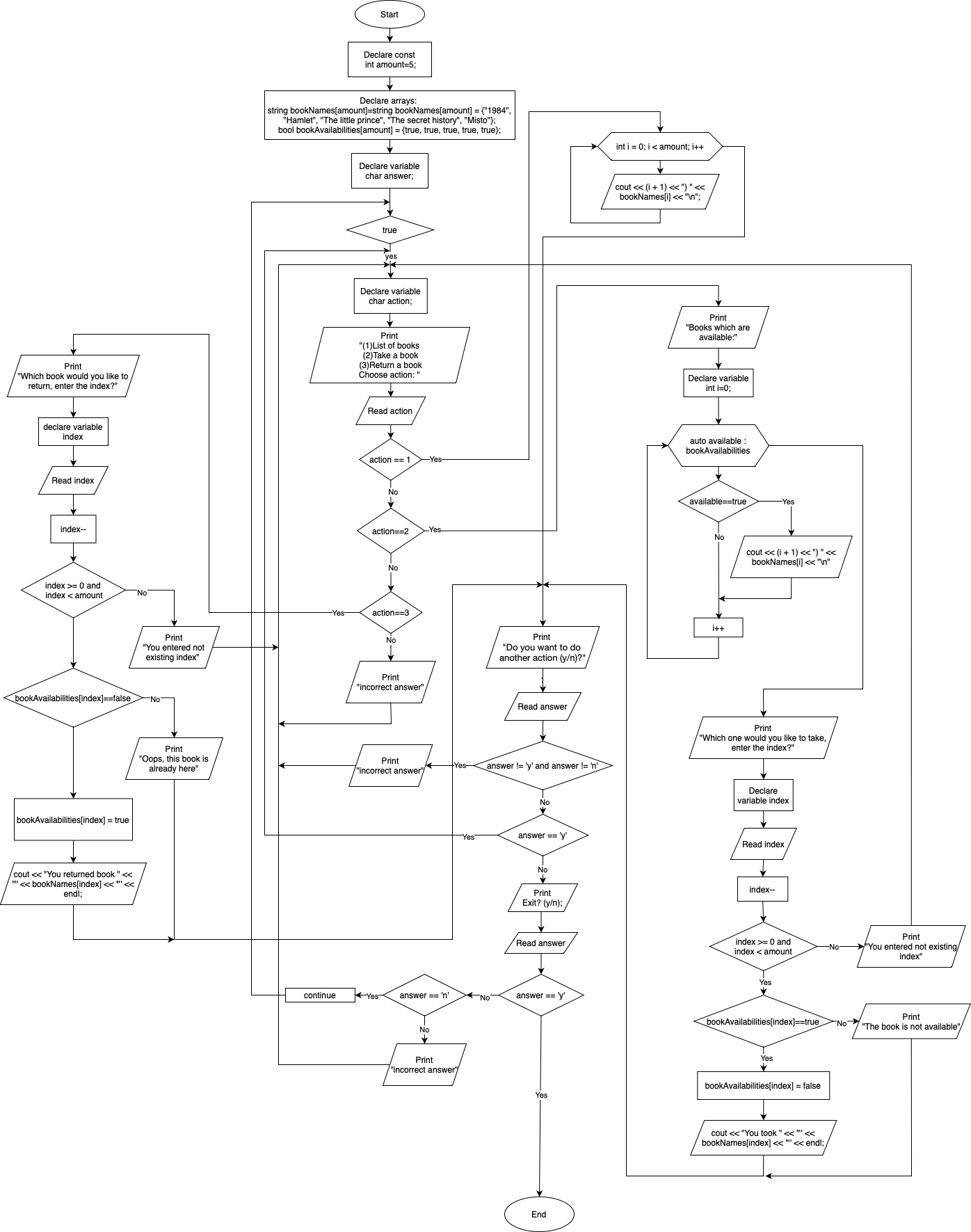
Програма №3 VNS Lab7 Task 1

* Блок-схема
* Планований час на реалізацію: 3 дні;

Програма №4 VNS Lab7 Task 2

* Планований час на реалізацію: 2 дні;

Програма №5 Class Practice Work «Менеджмент бібліотеки»



* Планований час на реалізацію: 2 дні;

Програма №6 Self Practice Task «Депутатські гроші»

* Планований час на реалізацію: 1 день

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

Завдання №1 VNS Lab 2

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_3_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%203/vns_lab_2_task_1_variant_16_olena_novosad.cpp>

#include <iostream>

#include <math.h>

#include <stdio.h>

using namespace std;

int factorial(int n)

{

    if (n == 0)

    {

        return 1;

    }

    else

    {

        return n \* factorial(n - 1);

    }

}

int main()

{

    double An;

    int n = 1;

    double A1 = log10(factorial(n)) \* exp((-n) \* pow(n, 0.5));

    n = 2;

    double A2 = log10(factorial(n)) \* exp((-n) \* pow(n, 0.5));

    double sum = A1 + A2;

    double eps = 0.0001;

    double prevA = A2;

    do

    {

        An = prevA \* ((log10(factorial(n + 1))) \* exp(-(n + 1) \* (pow((n + 1), 0.5)))) / ((log10(factorial(n))) \* exp(-n \* pow(n, 0.5)));

        prevA = An;

        sum += An;

        n++;

    } while (abs(An) > eps);

    cout << sum;

    return 0;

}

Рисунок : Код програми №1

Завдання №2 VNS Lab 3

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_3_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%203/vns_lab_3_task_1_variant_16_olena_novosad.cpp>

#include <iostream>

#include <math.h>

#include <stdio.h>

using namespace std;

int main()

{

    const double a = M\_PI / 5.0;

    const double b = M\_PI;

    const int k = 10;

    const int n = 40;

    const double eps = 0.0001;

    double krok = (b - a) / k;

    double x = M\_PI / 5.0;

    do

    {

        double SN = 0;

        double SE = 0;

        double se = 1;

        float eps = 0.0001;

        for (int i = 1; i <= n; i++)

        {

            double sn = (cos((2 \* i - 1) \* x)) / pow((2 \* i - 1), 2);

            SN = SN + sn;

        }

        int i;

        for (i = 1; abs(se) >= eps; i++)

        {

            se = (cos((2 \* i - 1) \* x)) / pow((2 \* i - 1), 2);

            SE += se;

        }

        double y = (pow(M\_PI, 2)) / 8 - (M\_PI / 4) \* abs(x);

        cout << "X=" << x << " "

             << "SN=" << SN << " "

             << "SE=" << SE << " "

             << "Y=" << y << " \n";

        x += krok;

    } while (x <= M\_PI);

    return 0;

}

Рисунок : Код до програми №2

Завдання №3 VNS Lab7 Task 1

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_3_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%203/vns_lab_7_task_1_variant_16_olena_novosad.cpp>

#include <iostream>

#include <cstdarg>

#include <iomanip>

#include <vector>

#include <sstream>

using namespace std;

int amountOfDaysBetweenDates(char \*date1, char \*date2)

{

    tm tm1 = {0};

    tm tm2 = {0};

    istringstream stream1(date1);

    stream1 >> get\_time(&tm1, "%d.%m.%y");

    time\_t time1 = mktime(&tm1);

    istringstream stream2(date2);

    stream2 >> get\_time(&tm2, "%d.%m.%y");

    time\_t time2 = mktime(&tm2);

    double d = difftime(time2, time1);

    return int(d) / (24 \* 60 \* 60);

}

void days(int k, ...)

{

    if (k < 2)

    {

        return;

    }

    vector<char\*> dates;

    va\_list args;

    va\_start(args, k);

    for (int i = 0; i < k; i++)

    {

        dates.push\_back(va\_arg(args, char \*));

    }

    va\_end(args);

    for (int a = 0; a < k - 1; a++)

    {

        cout << amountOfDaysBetweenDates(dates.at(a), dates.at(a + 1)) << " ";

    }

    cout << endl;

}

int main()

{

    days(2, "06.10.23", "18.10.24");

    days(4, "16.10.23", "18.10.23", "16.11.23", "18.11.23");

    days(7, "16.10.23", "18.10.23", "16.11.23", "18.11.23", "16.12.23", "18.12.23", "24.12.23");

    return 0;

}

Рисунок : Код до програми №3

Завдання №4 VNS Lab7 Task 2

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_3_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%203/vns_lab_7_task_2_variant_16_olena_novosad.cpp>

#include <iostream>

using namespace std;

struct Time{

int minutes;

int hours;

};

int convertTime(Time time)

{

    return time.minutes + time.hours \* 60;

}

Time convertTime(int minutes)

{

    Time time={0};

   time.hours=minutes/60;

   time.minutes=minutes%60;

   return time;

}

int main(){

    int time1 = 567;

    Time ttt =  convertTime(time1);

    cout << ttt.hours << " hours " << ttt.minutes << " minutes \n";

    Time ggg={0};

    ggg.hours=4;

    ggg.minutes=59;

    cout << convertTime(ggg) << " minutes \n";

}

Рисунок : Код до програми №4

Завдання №5 Class Practice Work «Менеджмент бібліотеки»

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_3_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%203/practice_work_task_1_olena_novosad.cpp>

#include <iostream>

#include <string>

using namespace std;

const int amount = 5;

string bookNames[amount] = {"1984", "Hamlet", "The little prince", "The secret history", "Misto"};

bool bookAvailabilities[amount] = {true, true, true, true, true};

int main()

{

    char answer;

    while (true)

    {

         do

        {

        start:

            char action;

            printf(" (1)List of books\n (2)Take a book\n (3)Return a book\n Сhoose action: ");

            cin >> action;

            if (action == '1')

            {

                for (int i = 0; i < amount; i++)

                {

                    cout << (i + 1) << ") " << bookNames[i] << "\n";

                }

            }

            else if (action == '2')

            {

                cout << "Books which are available: \n";

                int i = 0;

                for (auto available : bookAvailabilities)

                {

                    if (available)

                    {

                        cout << (i + 1) << ") " << bookNames[i] << "\n";

                    }

                    i++;

                }

                cout << "\nWhich one would you like to take, enter the index?\n";

                int index;

                cin >> index;

                index--;

                if (index >= 0 && index < amount)

                {

                    if (bookAvailabilities[index])

                    {

                        bookAvailabilities[index] = false;

                        cout << "You took " << '"' << bookNames[index] << '"' << endl;

                    }

                    else

                    {

                        cout << "The book is not available\n";

                    }

                }

                else

                {

                    cout << "You entered not exsisting index\n\n";

                    goto start;

                }

            }

            else if (action == '3')

            {

                cout << "Which book would you like to return, enter the index?\n";

                int index;

                cin >> index;

                index--;

                if (index >= 0 && index < amount)

                {

                    if (!bookAvailabilities[index])

                    {

                        bookAvailabilities[index] = true;

                        cout << "You returned book " << '"' << bookNames[index] << '"' << endl;

                    }

                    else

                    {

                        cout << "Oops, this book is already here\n";

                    }

                }

                else

                {

                    cout << "You entered not existing index\n\n";

                    goto start;

                }

            }

            else {

                cout << "incorrect answer\n\n";

                goto start;

            }

            cout << "Do you want to do another action (y/n)?\n";

            cin >> answer;

            if (answer != 'y' && answer != 'n') {

                cout << "incorrect answer\n\n";

                goto start;

            }

        } while (answer == 'y');

        cout << "Exit? (y/n)\n";

        cin >> answer;

        if (answer == 'y')

        {

            break;

        }

        else if (answer == 'n') {

            continue;

        }

        else {

            cout << "incorrect answer\n\n";

            goto start;

        }

    }

    return 0;

}

Рисунок : Код до програми №5

Завдання №6 Self Practice Task «Депутатські гроші»

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_3_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%203/self_practice_work_algotester_task_1_olena_novosad.cpp>

#include <iostream>

using namespace std;

int main()

{

    long int n;

    cin >> n;

    int amount = 0;

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

    int i = 0;

    while (n != 0)

    {

        if (n >= groshi[i])

        {

            amount += n / groshi[i];

            n = n % groshi[i];

        }

        i++;

    }

    cout << amount;

    return 0;

}

Рисунок : Код до програми №6

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

Завдання №1 VNS Lab 2

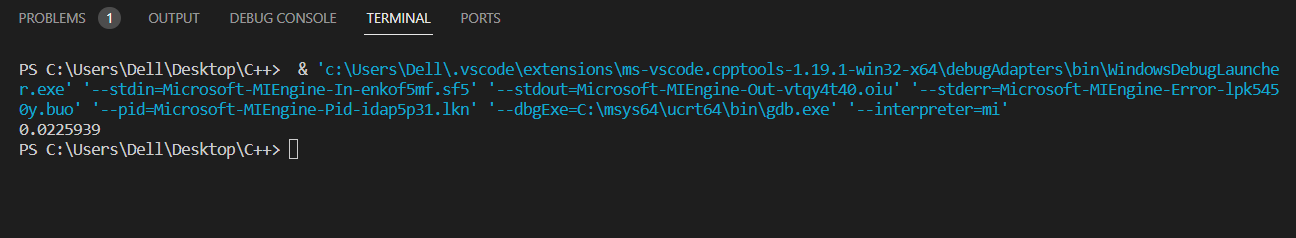


Рисунок : Результат тестування програмного коду №1

Час затрачений на виконання завдання: 2 дні

Завдання №2 VNS Lab 3

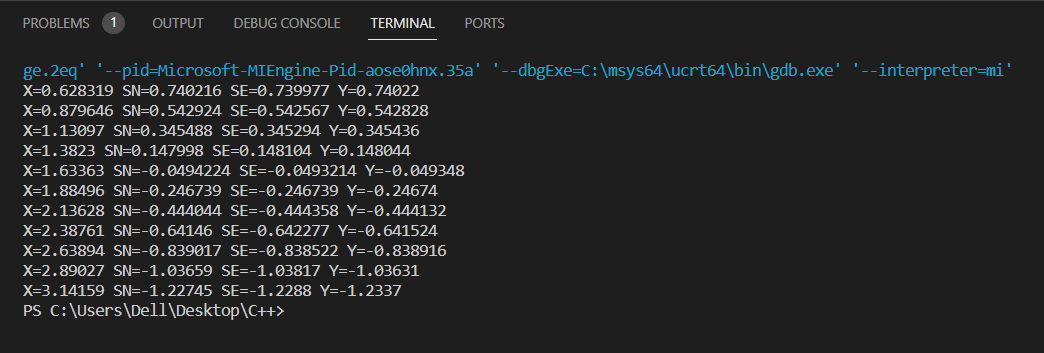


Рисунок : Результат тестування програмного коду №2

Час затрачений на виконання завдання: 2 дні

Завдання №3 VNS Lab7 Task 1

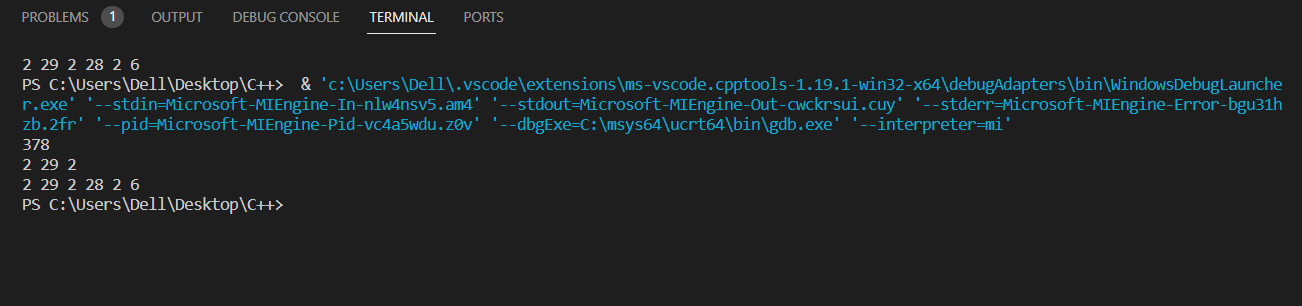


Рисунок : Результат тестування програмного коду №3

Час затрачений на виконання завдання: 4 дні

Завдання №4 VNS Lab7 Task 2

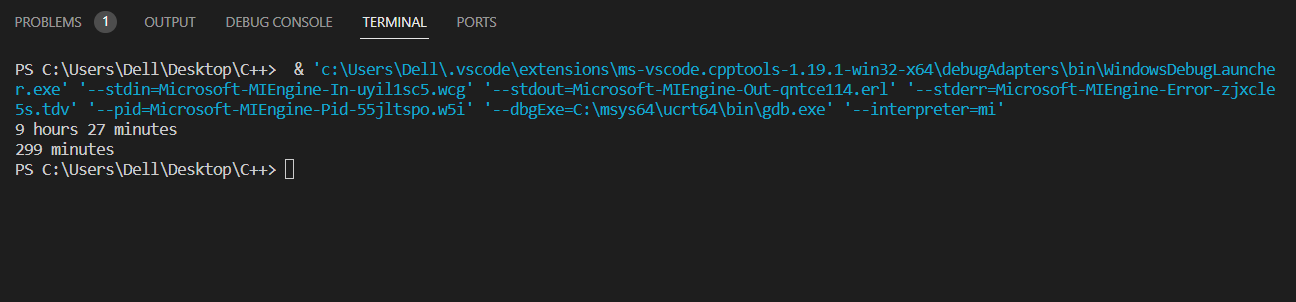


Рисунок : Результат тестування програмного коду №4

Час затрачений на виконання завдання: 1 день

Завдання №5 Class Practice Work «Менеджмент бібліотеки»

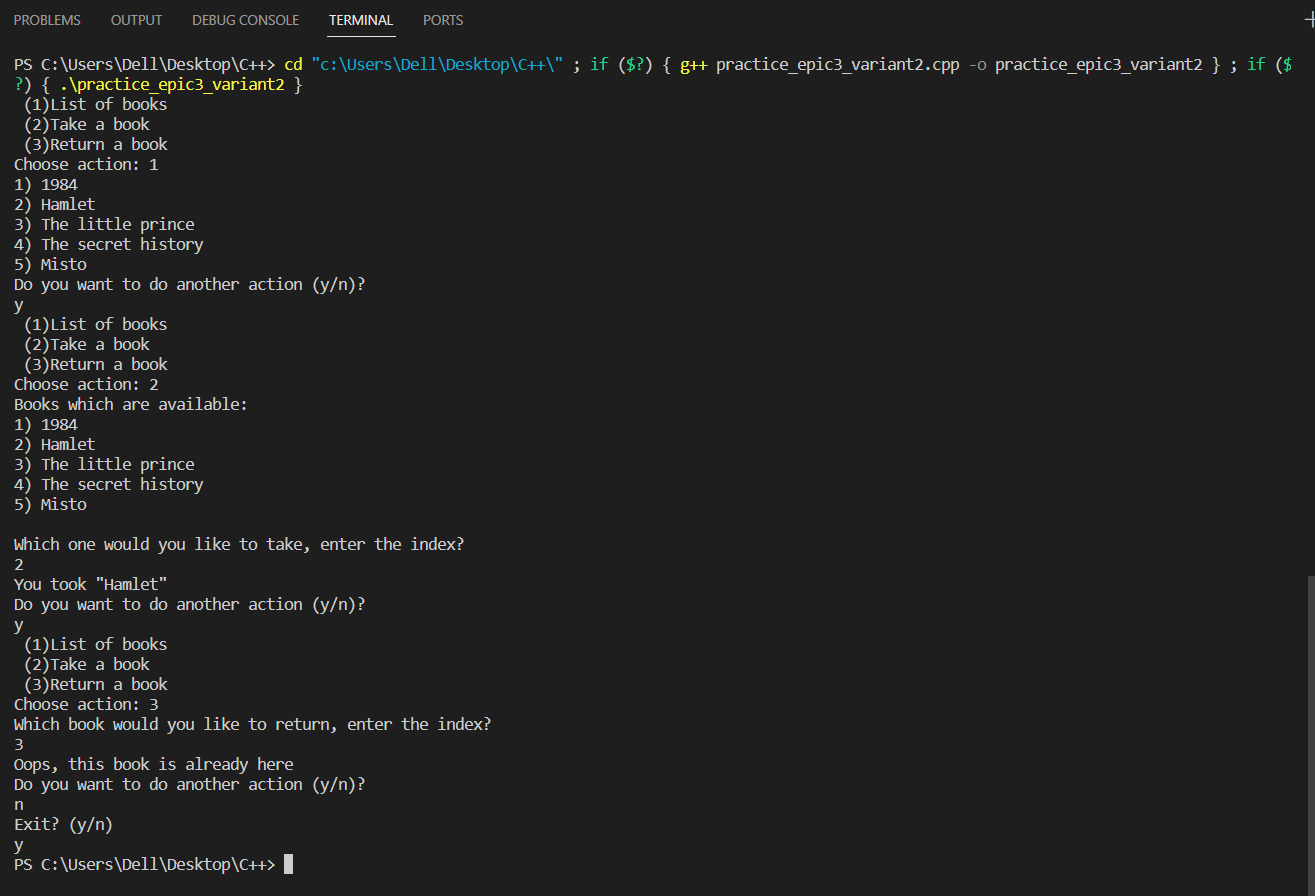


Рисунок : Результат тестування програмного коду №5

Час затрачений на виконання завдання: 3 дні

Завдання №6 Self Practice Task «Депутатські гроші»

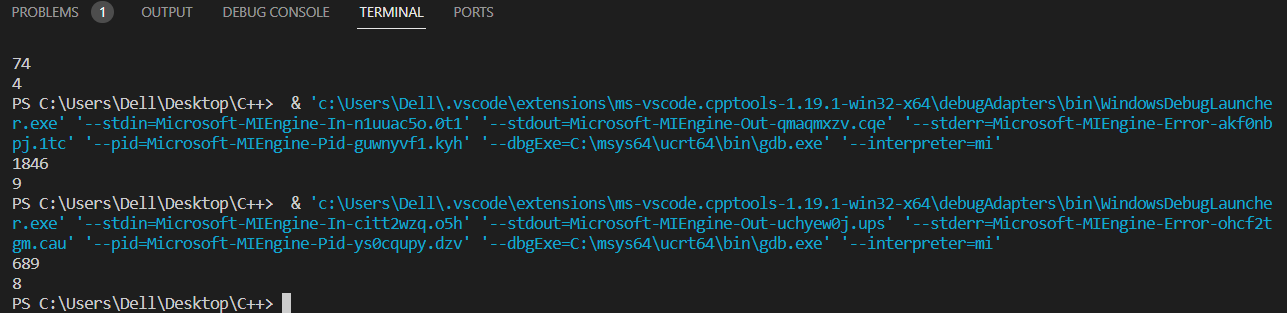


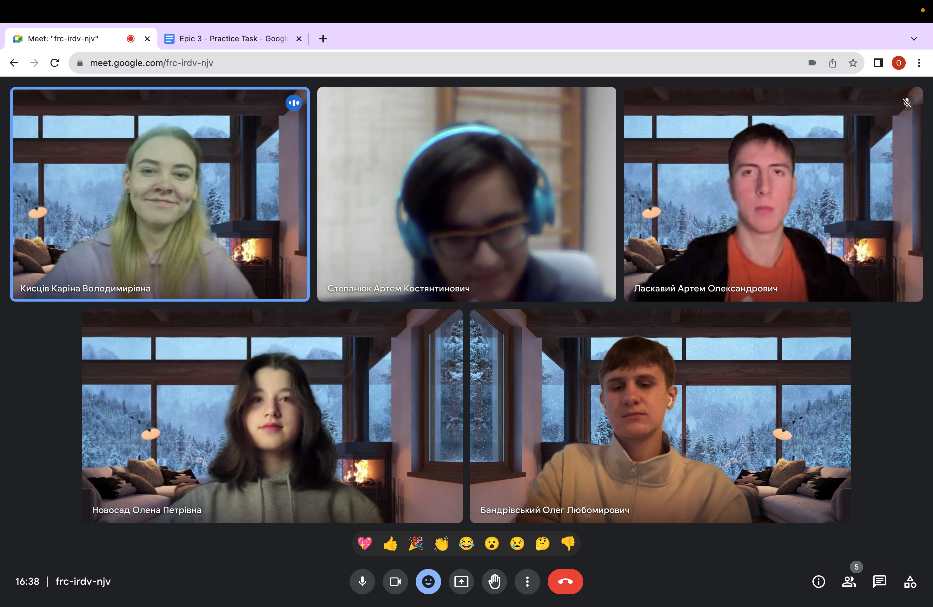
Рисунок : Результат тестування програмного коду №6



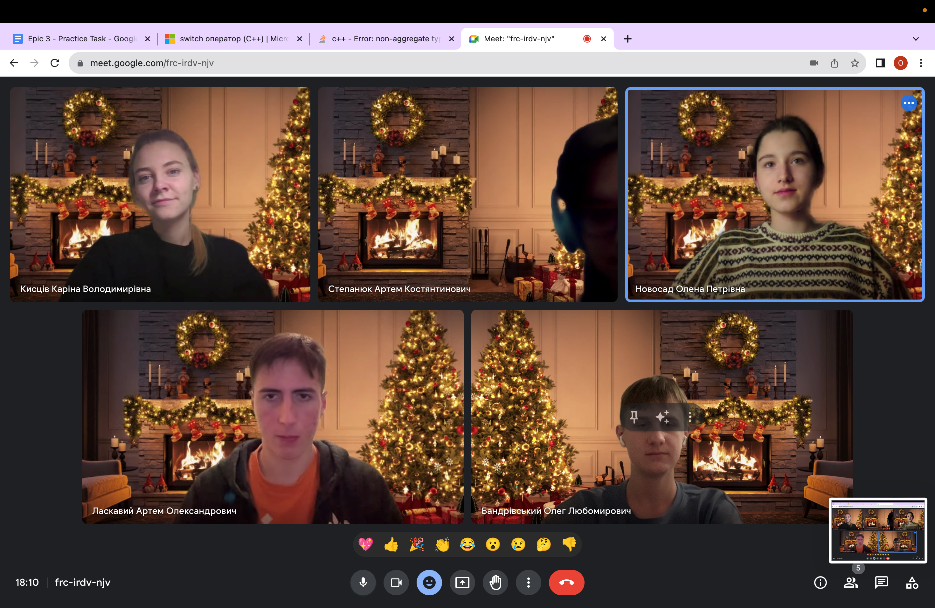
Час затрачений на виконання завдання: 1 день

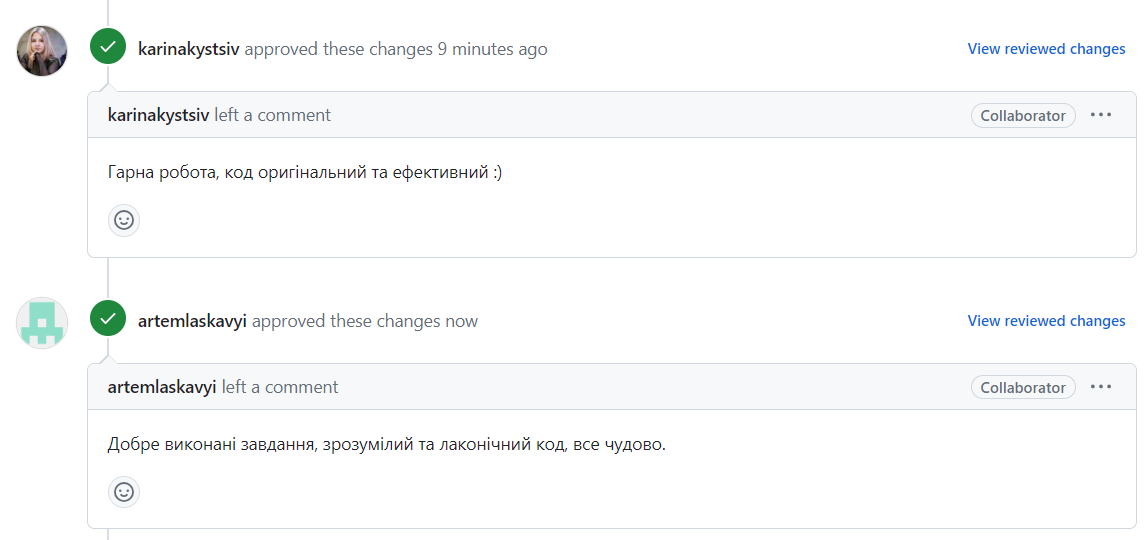
## **5. Кооперація з командою:**

Скрін з 1-ї зустрічі по обговоренню задач Епіку 3



Скрін з 2-ї зустрічі по обговоренню задач Епіку 3



* Скрін з 2-му коментарями від учасників команди на пул реквесті
* ****

# **Висновки:**

Під час виконання завдань та огляду теоретичного матеріалу було розглянуто різні аспекти роботи з функціями та їх перевантаженням, що дозволяє використовувати одне ім'я функції для виконання різних алгоритмів.

Був проведений аналіз інформації щодо видів циклів, таких як for, while, do while, for each, а також про оператори break, continue, goto; засвоєно на практиці застосування вкладених циклів при написанні програмного коду.

Окремо було приділено увагу рекурсії – концепції, де функція викликає сама себе. Для кращого розуміння було застосовано для створення факторіалу, необхідного для реалізації однієї з програм.

Після ознайомлення з інформацією про зазначені концепції було створено вище зазначені програми, використовуючи набуті знання. Це сприяло глибшому засвоєнню матеріалу та розвитку практичних навичок у написанні програмного коду.