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»

***Виконав:***

студент групи ШІ-14

Шелеп Андрій Іванович

# **Тема роботи:**

Цикли, використання циклів у С++. Вкладені цикли. Обчислення суми рядів за допомогою циклів. Функції та їх перевантаження. Рекурсія функції. Створення функції для обчислення виразу та функції зі змінною кількістю параметрів.

# **Мета роботи:**

Ознайомитися, використати та опрацювати цикли for, while, do while у С++. Ознайомитися та опрацювати функції та їхню організацію, перевантаження функції та рекурсію.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли. Вкладені цикли
* Тема №2: Оператори continue, break, goto.
* Тема №3: Функція.
* Тема №4: Перевантаження функції.
* Тема №5: Рекурсія

1. Індивідуальний план опрацювання теорії:

* Тема №1:Цикли. Вкладені цикли .
  + Джерела Інформації
    - Книжка. Мова програмування С. Браян В. Кернiган, Денiс М. Рiчi
    - Відео.

<https://www.youtube.com/watch?v=ckJtOMcIxyU&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>

<https://www.youtube.com/watch?v=QXaSSIjVor8&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>

<https://www.youtube.com/watch?v=pBhaBdXWMmU&t=575s&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>

* + - Стаття.

<http://cpp.dp.ua/operatory-tsyklu/>

<https://acode.com.ua/urok-72-tsykl-for/>

<https://acode.com.ua/urok-70-tsykl-while/>

* + Що опрацьовано:
    - Цикли у С++: for, while, do while. Різницю між цими циклами. Вкладені цикли
  + Статус: Ознайомлений
  + Початок опрацювання теми: 07.11.2023
  + Звершення опрацювання теми: 20.11.2023
* Тема №2: Оператори continue, break, goto.
  + Джерела Інформації:
    - Відео.

<https://www.youtube.com/watch?v=rj1OLsBKazA&t=399s&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>

<https://www.youtube.com/watch?v=UY295pIdeoQ&t=337s&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>

* + - Стаття.

<https://acode.com.ua/urok-73-operatory-break-i-continue/#toc-1>

<https://acode.com.ua/urok-69-operator-goto/>

* + Що опрацьовано:
    - Оператори переходу break, goto, continue та як вони застосовуються в циклах
  + Статус: Ознайомлений
  + Початок опрацювання теми: 07.11.2023
  + Звершення опрацювання теми: 20.11.2023
* Тема №3:Функція.
  + Джерела Інформації:
    - Відео.

<https://www.youtube.com/watch?v=G8P6SvdqU9s&t=1663s&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>

* + - Стаття.

<https://acode.com.ua/urok-15-funktsiyi-i-operator-return/>

* + Що опрацьовано:
    - Функції С++, типи повернення функцій, прототипи і визначення функції
  + Статус: Ознайомлений
  + Початок опрацювання теми: 14.11.2023
  + Звершення опрацювання теми: 22.11.2023
* Тема №4 Перевантаження функції.
  + Джерела Інформації:
    - Відео.

<https://www.youtube.com/watch?v=3KJfisev6SI&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>

* + - Стаття.

<https://acode.com.ua/urok-108-perevantazhennya-funktsij/>

* + Що опрацьовано:
    - Перевантаження функції
  + Статус: Ознайомлений
  + Початок опрацювання теми: 14.11.2023
  + Звершення опрацювання теми: 22.11.2023
* Тема №5 Рекурсія.
  + Джерела Інформації:
    - Відео.

<https://www.youtube.com/watch?v=V7q9w_s0nns&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>

* + - Стаття.

<https://acode.com.ua/urok-113-rekursiya-i-chysla-fibonachchi/>

<http://cpp.dp.ua/rekursyvni-funktsiyi/>

* + Що опрацьовано:
    - Що таке рекурсія. Рекурсивні функції
  + Статус: Ознайомлений
  + Початок опрацювання теми: 14.11.2023
  + Звершення опрацювання теми: 22.11.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 2 - Task 1

* Варіант 14
* Використовуючи оператор циклу, знайти суму ряду, загальний член якого

, з точністю 0,0001

* Використання рекурентної формули для обчислення наступного елемента через попередній та використання циклу для обчислення суми ряду.

Завдання №2 VNS Lab 3 - Task 1

* Варіант 14
* Обчислити функцію f(x), використовуючи її розклад в степеневий ряд у двох випадках:

а) для заданого n = 20;

б) для заданої точності ε (ε=0.0001).



* Алгоритм розв’язання завдання зводиться до трьох циклів, причому два з

них вкладені в третій. Підключення заголовочного файлу <cmath>

Завдання №3 VNS Lab 7 - Task 1

* Варіант 14
* Написати функцію зі змінною кількістю параметрів для перетворення чисел

із двійкової системи числення в трійкову.

* Написано функцію зі змінною кількістю параметрів. Викликати функцію не менше трьох разів з кількістю параметрів 3, 6, 7.

Завдання №4 VNS Lab 7 - Task 2

* Варіант 14
* Написати перевантажені функції й основну програму, що їх викликає.

а) для обчислення натурального логарифму;

б) для обчислення десяткового логарифму.

* Організувано перевантаження функції

Завдання №5 Class Practice Work

* Без варіантів
* Проста програма керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.
* Використати наступні оператори:

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 each: перевірити наявність кожної книги.

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

Завдання №6 Self Practice Work

* Задача з алготестеру 0531 - Веселі каруселі
* Марічка має k гривень, а в парку є n атракціонів. Для кожного атракціону відома його ціна c[j]. Вам необхідно визначити кількість атракціонів, на яких покатається Марічка.

Марічка розпочне з найдешевшого атракціону, потім покатається на другому найдешевшому і так далі, доки в неї вистачить грошей

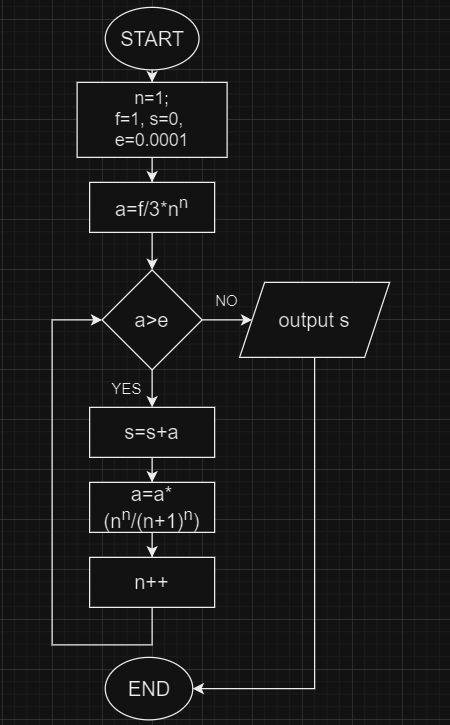
Завдання №7 Self Practice Work

* Задача з алготестеру 1792 - Стипендія.
* Якщо в студента всі оцінки відмінні (не менші за 90), то йому призначається підвищена стипендія. Якщо всі оцінки студента не менші за 51, але він не має права на підвищену стипендію, він отримує звичайну стипендію. Якщо студент має хоч одну незадовільну оцінку (меншу за 51), йому стипендія не призначається. Дізнатися, яка стипендія у студента
* Використовував цикл

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

Програма №1 VNS Lab 2 - Task 1

* Блок-схема

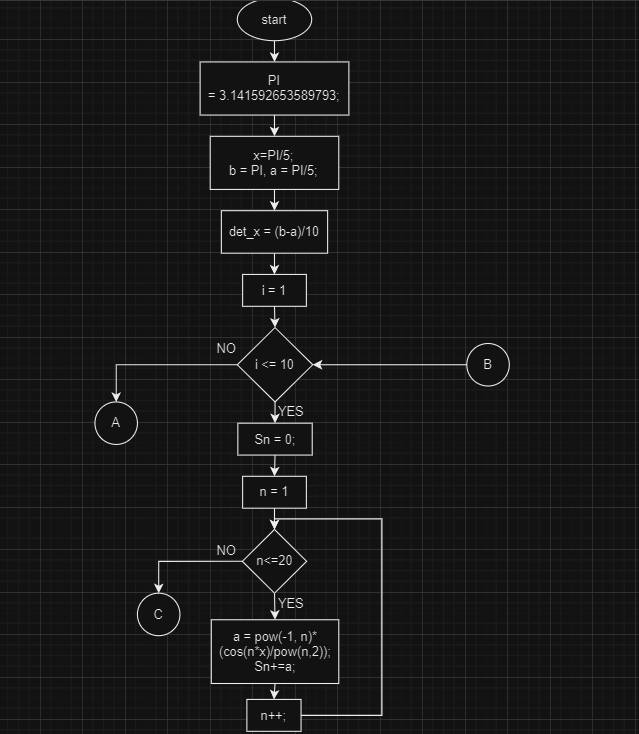


*Image 1**. Блок-схема до програми №1 VNS Lab 2 Task 1*

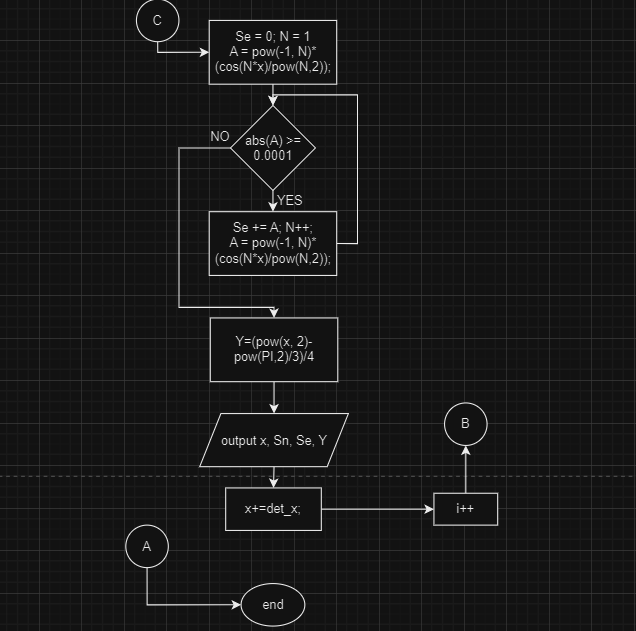
* Планований час на реалізацію: 2 год
* Важливі деталі для врахування в імплементації. Використання рекурентної формули для обчислення наступного елемента через попередній. Використано цикл while для обчислення суми ряду

Програма №2 VNS Lab 3 - Task 1

* Блок-схема



*Image 2. Блок-схема до програми №2 VNS Lab 3 Task 1*



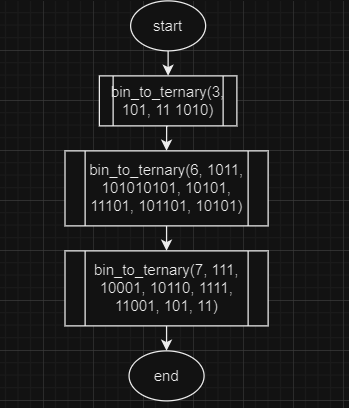
*Image 3. Блок-схема до програми №2 VNS Lab 3 Task 1*

* Планований час на реалізацію: 2 год
* Важливі деталі для врахування в імплементації. Три цикли, причому два з

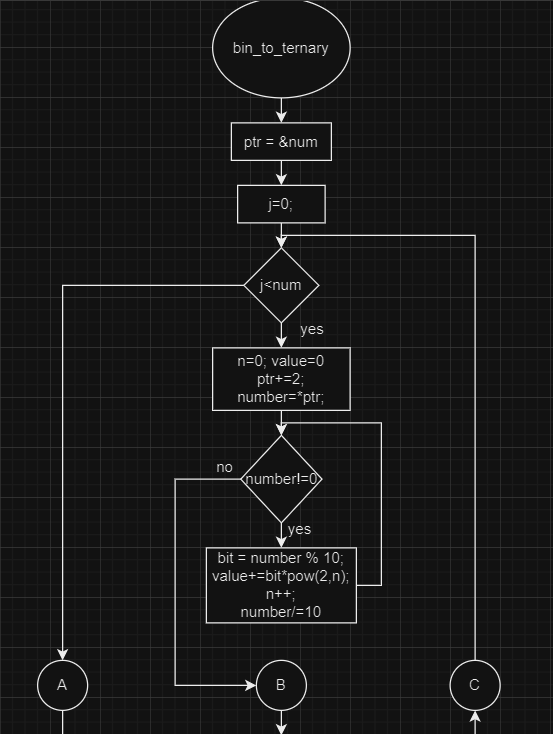
них вкладені в третій.

Програма №3 VNS Lab 7 - Task 1

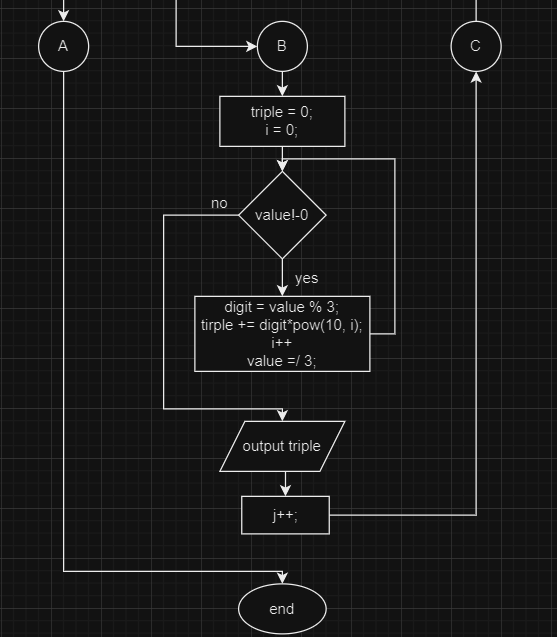
* Блок-схема



*Image 4. Блок-схема до програми №3 VNS Lab 7 Task 1*



*Image 5. Блок-схема до програми №3 VNS Lab 7 Task 1*

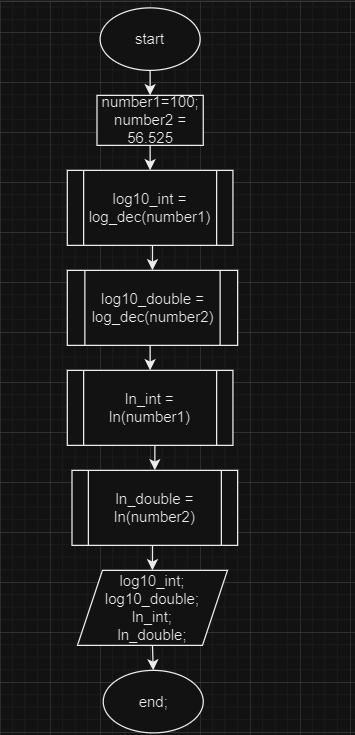


*Image 6. Блок-схема до програми №3 VNS Lab 7 Task 1*

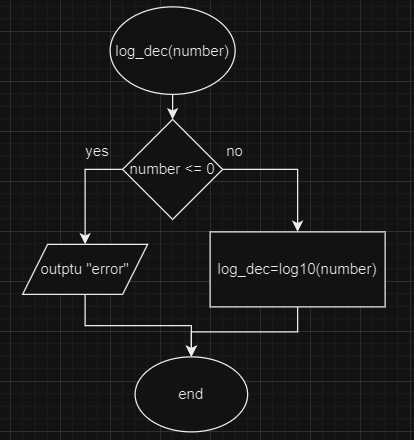
* Планований час на реалізацію: 30 хв
* Важливі деталі для врахування в імплементації. Написано функцію зі змінною кількістю параметрів. Викликано функцію не менше трьох разів з кількістю параметрів 3, 6, 7.

Програма №4 VNS Lab 7 - Task 2

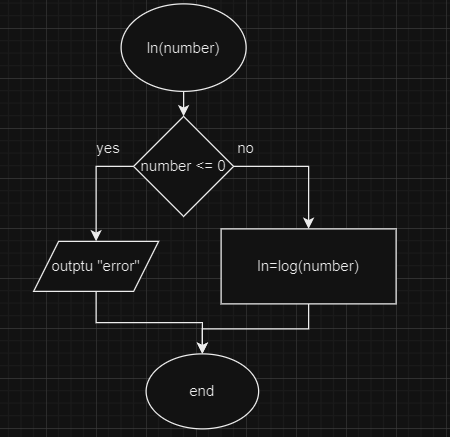
* Блок-схема



*Image 7. Блок-схема до програми №4 VNS Lab 7 Task 2*



*Image 8. Блок-схема до програми №4 VNS Lab 7 Task 2*

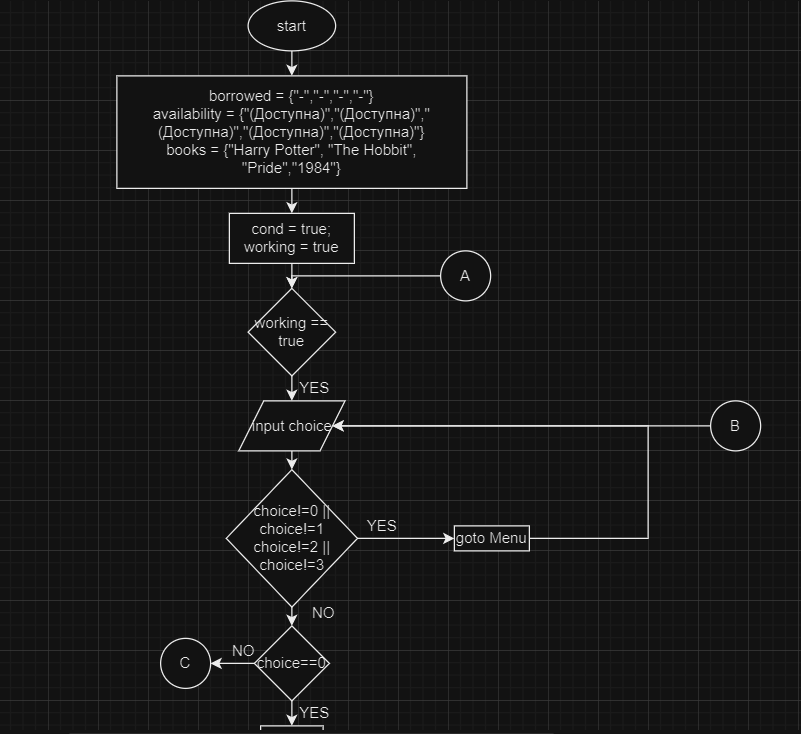


*Image 9. Блок-схема до програми №4 VNS Lab 7 Task 2*

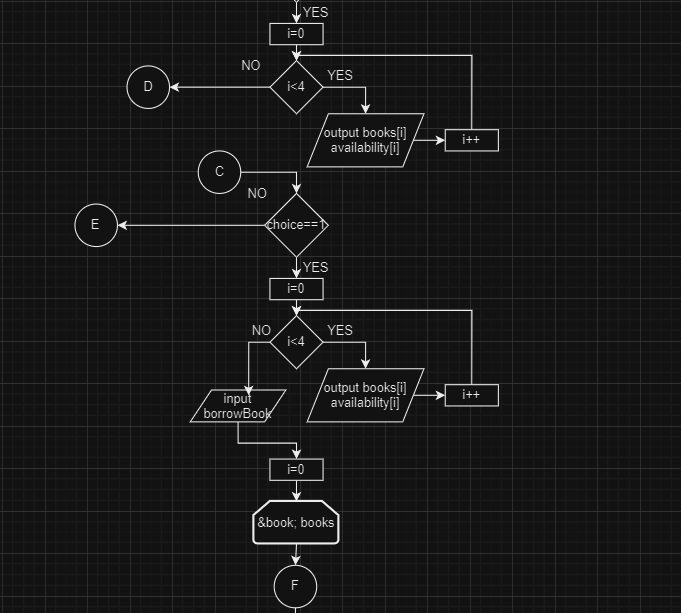
* Планований час на реалізацію: 2 дні
* Важливі деталі для врахування в імплементації. Організувано перевантажені функції

Програма №5 Class Practice Work

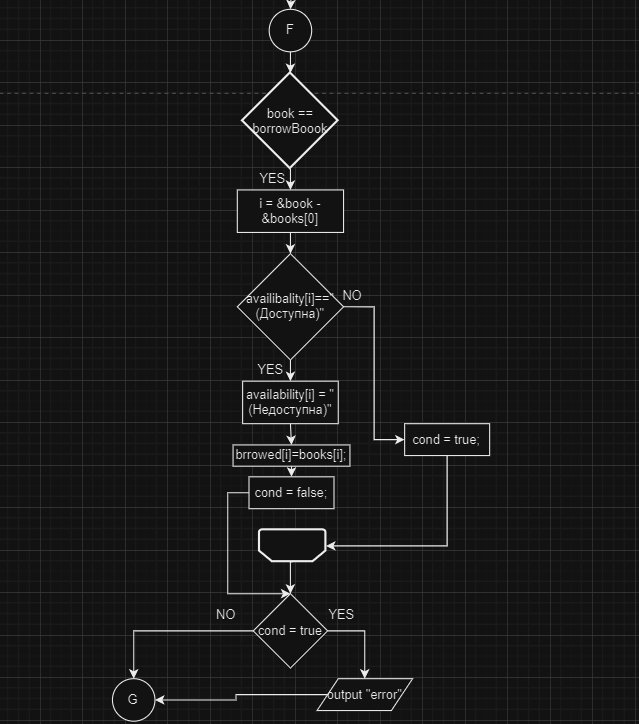
* Блок-схема



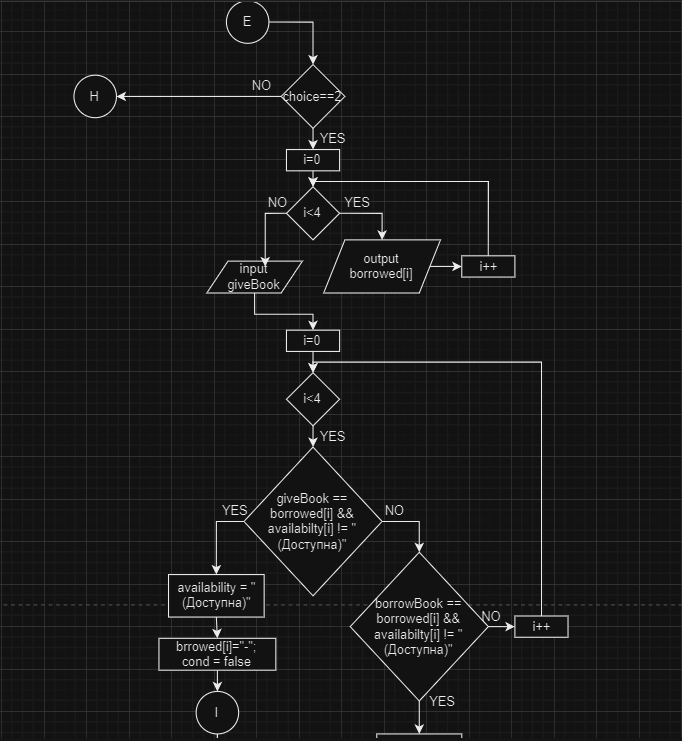
*Image 10. Блок-схема до програми №5 Class Practice Work*



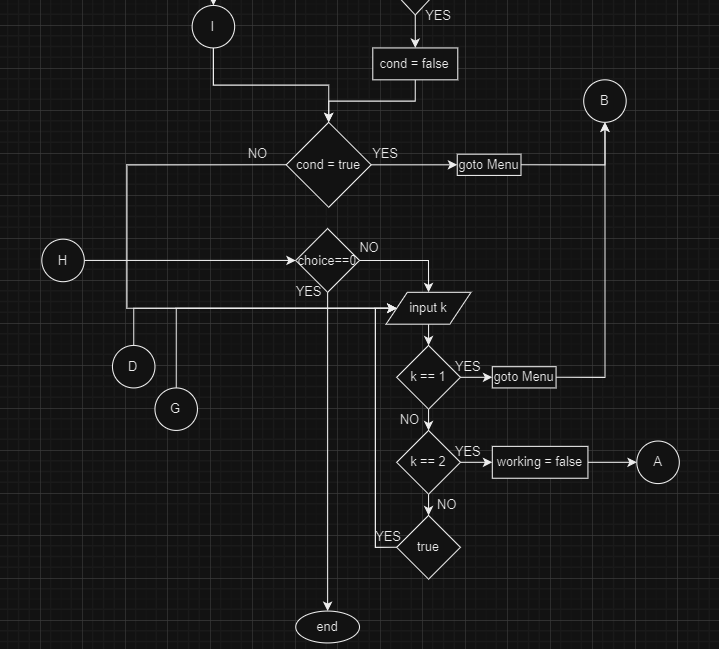
*Image 11. Блок-схема до програми №5 Class Practice Work*



*Image 12. Блок-схема до програми №5 Class Practice Work*



*Image 13. Блок-схема до програми №5 Class Practice Work*



*Image 14. Блок-схема до програми №5 Class Practice Work*

* Планований час на реалізацію: 3 дні
* Важливі деталі для врахування в імплементації. 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.

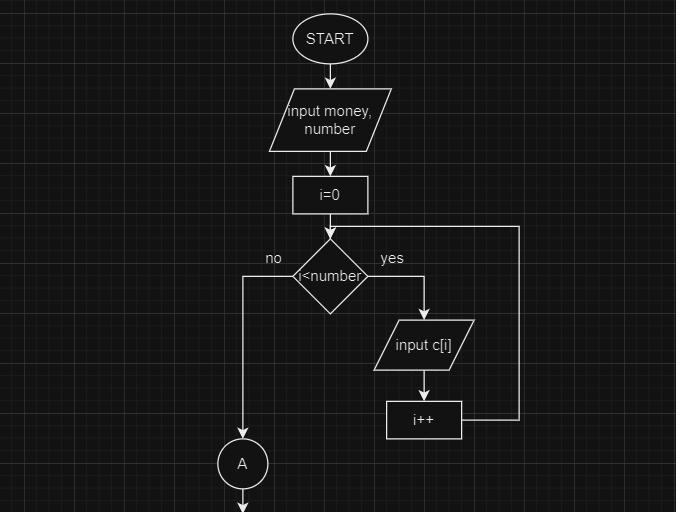
for: список усіх книг за допомогою циклу.

for each: перевірити наявність кожної книги.

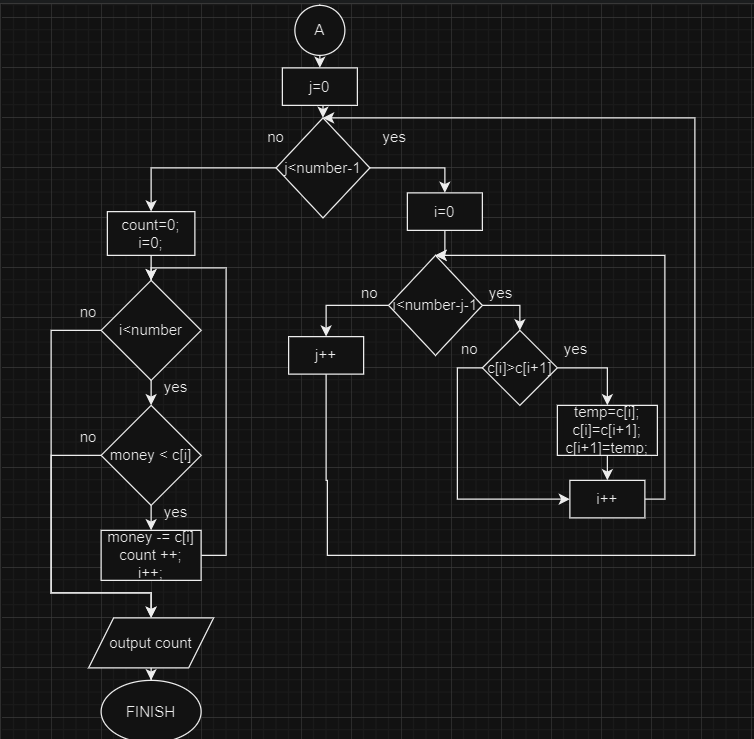
goto: якщо користувач вводить неправильний вибір, використовуйте goto, щоб перенаправити його до головного меню.

Програма №6 Self Practice Work

* Блок-схема



*Image 15. Блок-схема до програми №6 Self Practice Work*

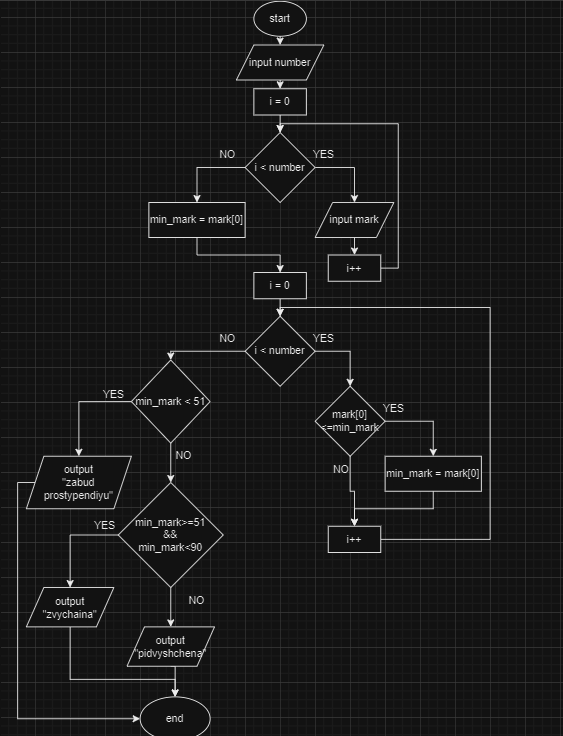


*Image 16. Блок-схема до програми №6 Self Practice Work*

* Планований час на реалізацію 1 год
* Важливі деталі для врахування в імплементації.

Програма №7 Self Practice Work

* Блок-схема



*Image 17. Блок-схема до програми №7 Self Practice Work*

* Планований час на реалізацію 15 хв

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

Використано налаштування з попередніх лабораторних робіт

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

Посилання на файл програми у пул-запиті GitHub: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/509>

Завдання №1 VNS Lab 2 - Task 1

Використовуючи оператор циклу, знайти суму ряду з точністю 0,0001

#include <iostream>

#include <math.h>

using *namespace* std;

*int* main(){

*int* n = 1;

*double* a, f=1, sum=0, e=0.0001;

a = f/(3\*pow(n,n));

while (a >= e)

{

sum += a;

a = a\*(pow(n,n)/pow(n+1,n));

n++;

}

cout << "sum: " << sum << endl;

return 0;

}

*Блок №1. Код до програми №1 VNS Lab 2 - Task 1*

*#include <iostream>*

*#include <math.h>*

*using namespace std;*

*int main(){*

*int n = 1;*

*double a=1, f=1, s=0, e=0.0001;*

*while (a >= e)*

*{*

*f\*=n;*

*a = f/(3\*pow(n,n));*

*s += a;*

*n++;*

*}*

*cout << "sum: " << s << endl;*

*return 0;*

*}*

*Блок №2. Код до програми №1 VNS Lab 2 - Task 1*

Завдання №2 VNS Lab 3 - Task 1

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

а) для заданого n = 20;

б) для заданої точності ε (ε=0.0001).

#include <stdio.h>

#include <math.h>

using *namespace* std;

*int* main(){

const *double* PI = 3.141592653589793238;

*int* n, N;

*double* Y, A, A1, x = PI/5, a = PI/5, b = PI, Sn, Se;

*double* det\_x = (b-a)/10;

for(*int* i = 1; i <= 10; i++){

Sn = 0;

for (n = 1; n <= 20; n++){

A = pow(-1,n)\*(cos(n\*x)/pow(n,2));

Sn += A;

}

Se = 0;

N = 1;

A1 = pow(-1, N) \* (cos(N \* x) / pow(N, 2));

while (abs(A1) >= 0.0001){

Se += A1;

N++;

A1 = pow(-1, N) \* (cos(N \* x) / pow(N, 2));

}

Y = (pow(x,2)-pow(PI,2)/3)/4;

printf("X=%lf\t SN=%lf\t SE=%lf\t Y=%lf\t\n", x, Sn, Se, Y);

x += det\_x;

}

return 0;

}

*Блок №3. Код до програми №2 VNS Lab 3 - Task 1*

Завдання №3 VNS Lab 7 - Task 1

Написати функцію зі змінною кількістю параметрів для перетворення чисел із двійкової системи числення в трійкову.

#include <iostream>

#include <cmath>

using *namespace* std;

*void* bin\_to\_ternary(*long* *long* *num*,...){

*long* *long* \* ptr = &*num*;

*int* n, value, bit, triple, digit, i, number;

for(*int* j = 0; j<*num*; j++){

n = 0;

ptr++;

number = \*ptr;

value = 0;

while(number!=0){

bit = number % 10;

value += bit\*pow(2, n);

n++;

number /= 10;

}

triple = 0;

i = 0;

while(value != 0){

digit = value % 3;

triple += digit\*pow(10,i);

i++;

value /= 3;

}

cout << triple << " ";

}

cout << endl;

}

*int* main(){

bin\_to\_ternary(3, 101, 11, 1010);

bin\_to\_ternary(6, 1011, 101010101, 10101, 11101, 101101, 10101);

bin\_to\_ternary(7, 111, 10001, 10110, 1111, 11001, 101, 11);

return 0;

}

*Блок №4. Код до програми №3 VNS Lab 7 - Task 1*

Завдання №4 VNS Lab 7 - Task 2

Написати перевантажені функції й основну програму, що їх викликає.

а) для обчислення натурального логарифму;

б) для обчислення десяткового логарифму.

#include <iostream>

#include <cmath>

using *namespace* std;

*double* log\_dec(*int* *number*){

if(*number* <= 0){

cout << "error" << endl;

return 0;

} else{

return log10(*number*);

}

}

*double* log\_dec(*double* *number*){

if(*number* <= 0){

cout << "error" << endl;

return 0;

} else{

return log10(*number*);

}

}

*double* ln(*int* *number*){

if(*number* <= 0){

cout << "error" << endl;

return 0;

} else{

return log(*number*);

}

}

*double* ln(*double* *number*){

if(*number* <= 0){

cout << "error" << endl;

return 0;

} else{

return log(*number*);

}

}

*int* main(){

*int* number1 = 100;

*double* number2 = 56.525;

*double* log10\_int = log\_dec(number1);

*double* log10\_double = log\_dec(number2);

*double* ln\_int = ln(number1);

*double* ln\_double = ln(number2);

cout << log10\_int << endl;

cout << log10\_double << endl;

cout << ln\_int << endl;

cout << ln\_double << endl;

return 0;

}

*Блок №5. Код до програми №4 VNS Lab 3 - Task 2*

Завдання №5 Class Practice Work.

Проста програма керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

#include <iostream>

#include <vector>

#include <string>

using *namespace* std;

*int* main(){

*string* borrowBook, giveBook;

*vector*<*string*> borrowed = {"-", "-", "-", "-"};

*vector*<*string*> availability = {"(Доступна)", "(Доступна)", "(Доступна)", "(Доступна)"};

*vector*<*string*> books = {"Harry Potter", "The Hobbit", "Pride and Prejudice", "1984"};

*int* choice;

*bool* cond = true;

*bool* working = true;

while (working == true){

Menu:

cout << "0 - Вийти \n1 - Позичити книгу \n2 - Повернути книгу \n3 - Перерахувати книги\n";

cin >> choice;

if (choice!= 0 && choice != 1 && choice != 2 && choice != 3){

cout << "виберіть ще раз" << endl;

goto Menu;

} else {

if (choice==3){

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

cout << "\n" << i+1 << " " << "\"" << books[i] << "\"" << " " << availability[i] << " " <<"\n";

}

} else if (choice == 1){

cout << "Виберіть книгу, яку хочете позичити: ";

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

cout << "\n" << i+1 << " " << "\"" << books[i] << "\"" << " " << availability[i] << " " <<"\n";

}

cin.ignore();

getline(cin, borrowBook);

for(const *auto* &book : books){

if (book == borrowBook){

*int* i = &book - &books[0];

if(availability[i]=="(Доступна)"){

cout << "ви взяли книгу " << "\"" << books[i] << "\"" << endl;

availability[i] = "(Недоступна)";

borrowed[i] = {books[i]};

} else{

cout << "Ця книга недоступна" << endl;

}

cond = false;

break;

} else{

cond = true;

}

}

if(cond == true){

cout << "Помилка" << endl;

}

} else if (choice == 2){

cout << "Яку книгу ви хочете повернути?";

for (*int* i = 0; i < 4; i++){

cout << "\n" << i+1 << " " << "\"" << borrowed[i] << "\"" << " " << " " <<"\n";

}

cin.ignore();

getline(cin, giveBook);

for(*int* i = 0; i < borrowed.size(); i++){

if (giveBook == borrowed[i] && availability[i]!="(Доступна)"){

cout << "ви повернули книгу " << "\"" << giveBook << "\"" << endl;

availability[i] = "(Доступна)";

borrowed[i] = "-";

cond = false;

break;

} else if(giveBook == borrowed[i] && availability[i]=="(Доступна)") {

cout << "Щось пішло не так" << endl;

cond = false;

break;

}

}

if (cond == true){

cout << "Помилка! Схоже ви ввели щось неправильно(" << endl;

goto Menu;

}

} else if (choice == 0){

cout << "bye";

break;

}

do{

cout << "Чи бажаєте ви виконати іншу операцію?\n1 - так\n2 - ні\n";

*int* k;

cin >> k;

if (k == 1){

goto Menu;

} else if (k == 2){

cout << "bye";

working = false;

break;

} else {

cout << "Спробуйте ще раз" << endl;

}

} while (true);

}

}

return 0;

}

*Блок №6. Код до програми №5 Class Practice Work*

Завдання №6 Self Practice Work

Марічка має k гривень, а в парку є n атракціонів. Для кожного атракціону відома його ціна c[j]. Вам необхідно визначити кількість атракціонів, на яких покатається Марічка.

// algotester 0531 веселі каруселі

#include <iostream>

using *namespace* std;

*int* main(){

*int* c[100], number , money;

cin >> money >> number;

for (*int* i = 0; i< number; i++){

cin >> c[i];

}

for (*int* j = 0; j < number - 1; j++){

for (*int* i = 0; i < number - j - 1; i++) {

if (c[i] > c[i+1]){

*int* temp = c[i];

c[i] = c[i+1];

c[i+1] = temp;

}

}

}

*int* count = 0;

for(*int* i = 0; i < number; i++){

if(money < c[i]){

break;

} else{

money-=c[i];

count++;

}

}

cout << count << endl;

return 0;

}

*Блок №7. Код до програми №6 Self Practice Work*

Завдання №6 Self Practice Work

Марічка має k гривень, а в парку є n атракціонів. Для кожного атракціону відома його ціна c[j]. Вам необхідно визначити кількість атракціонів, на яких покатається Марічка.

//задача з алготестеру 1792 - Стипендія

#include <iostream>

using *namespace* std;

*int* main(){

*int* number;

cin >> number;

*int* mark[7];

for(*int* i = 0; i < number; i++){

cin >> mark[i];

}

*int* min\_mark = mark[0];

for(*int* i = 0; i < number; i++){

if(mark[i]<=min\_mark)

min\_mark = mark[i];

}

if(min\_mark<51){

cout << "Zabud pro stypendiiu";

} else if(min\_mark>=51 && min\_mark < 90){

cout << "Zvychaina";

} else{

cout << "Pidvyshchena";

}

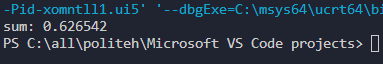
}

*Блок №8. Код до програми №7 Self Practice Work*

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

Завдання №1 VNS Lab 2 - Task 1

Програма обчислила суму ряду до того, поки наступний елемент не став менший за точність.

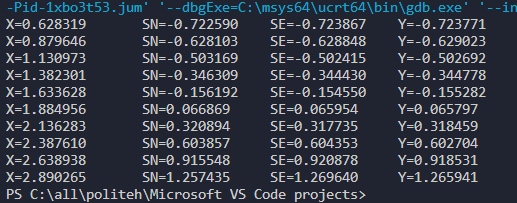


*Image 18**. Результати виконання завдання №1 VNS Lab 2 - Task 1*

Час затрачений на виконання завдання 1,5 год

Завдання №2 VNS Lab 3 - Task 1

Програма обчислила суми показникового ряду (при n=20, де n - к-ть елементів, і зі заданою точністю 0,0001) і точне значення функції при певному X.

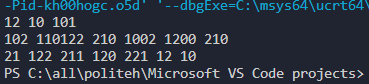


*Image 19. Результати виконання завдання №2 VNS Lab 3 - Task 1*

Час затрачений на виконання завдання 1,5 год

Завдання №3 VNS Lab 7 - Task 1

При викликах функції з різною кількістю параметрів, програма перетворює числа з двійкової системи в трійкову.

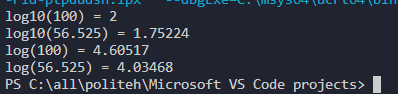


*Image 19. Результати виконання завдання №3 VNS Lab 7 - Task 1*

Час затрачений на виконання завдання 2 дні

Завдання №4 VNS Lab 7 - Task 2

Програма рахує десятковий та натуральний логарифми за допомогою перевантажених функцій для різних типів даних.

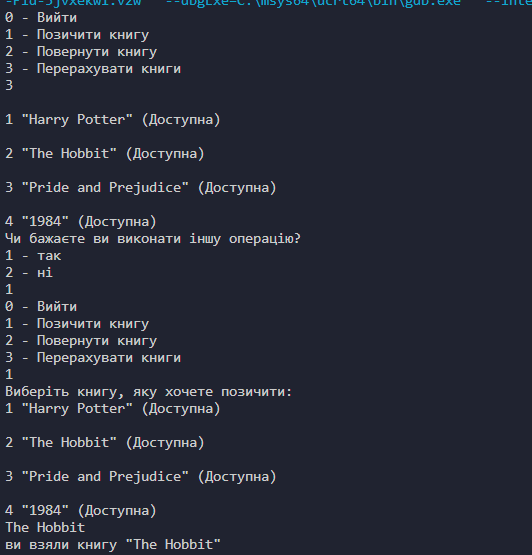


*Image 20. Результати виконання завдання №4 VNS Lab 7 - Task 2*

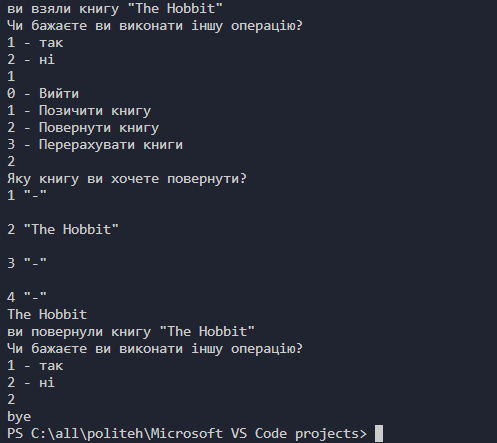
Час затрачений на виконання завдання 30 хв

Завдання №5 Class Practice Work

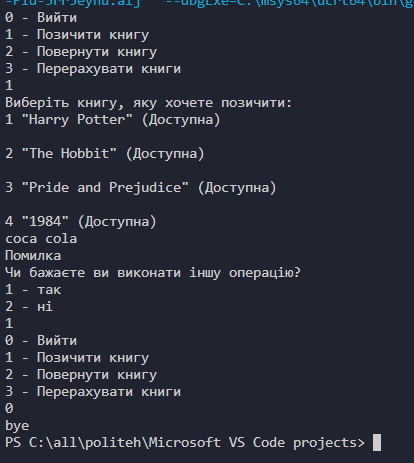
Спочатку попросив перерахувати книги. Програма запитує, чи хочу я виконати іншу дію. Я вибираю так і прошу позичити книгу. Програма дає мені список книг і їх статус доступності. Я вибираю книгу “The Hobbit”. Програма знову питає чи я хочу виконати іншу операцію. Далі я прошу повернути книгу і повертаю назад книгу “The Hobbit”. ПРограма запитує чи я хочу виконати іншу операцію. Я вибрав “ні” і програма закінчила роботу.



*Image 21. Результати виконання завдання №6 Class Practice Work*

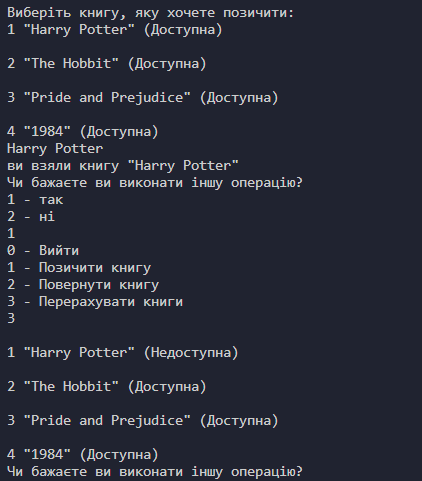


*Image 22. Результати виконання завдання №6 Class Practice Work*



*Image 22. Результати виконання завдання №6 Class Practice Work*

Тут я попробував взяти книгу, якої не існує. Програма вибила помилку і запитує, чи ми хочемо виконати іншу дію

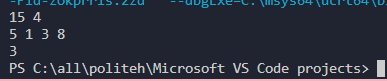


*Image 22. Результати виконання завдання №6 Class Practice Work*

Позичив книгу “Harry Potter”, і попросив ще раз перерахувати книгу. Видно що книга “Harry Potter” стала недоступною

Час затрачений на виконання завдання 3 дні

Завдання №6 Self Practice Work



*Image 23. Результати виконання завдання №6 Self Practice Work*

У Марічки було 15 грн та є 4 атракціони. Спочатку вона піде на атракціон за 1 гривню, потім за 3 гривні, а потім за 5 гривень, а на атракціон за 8 грн у неї не вистачить грошей. Сумарно Марічка відвідала 3 атракціони

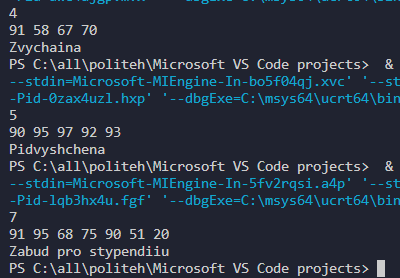
Час затрачений на виконання завдання 30 хв

Завдання №7 Self Practice Work

У першому випадку є бали нижчі за 90, тому у студента звичайна стипендія.

У другому випадку всі бали відмінні, тому у студента підвищена стипендія.

У третьому випадку є бал нижчий за 51, тому у студента стипендії не буде.

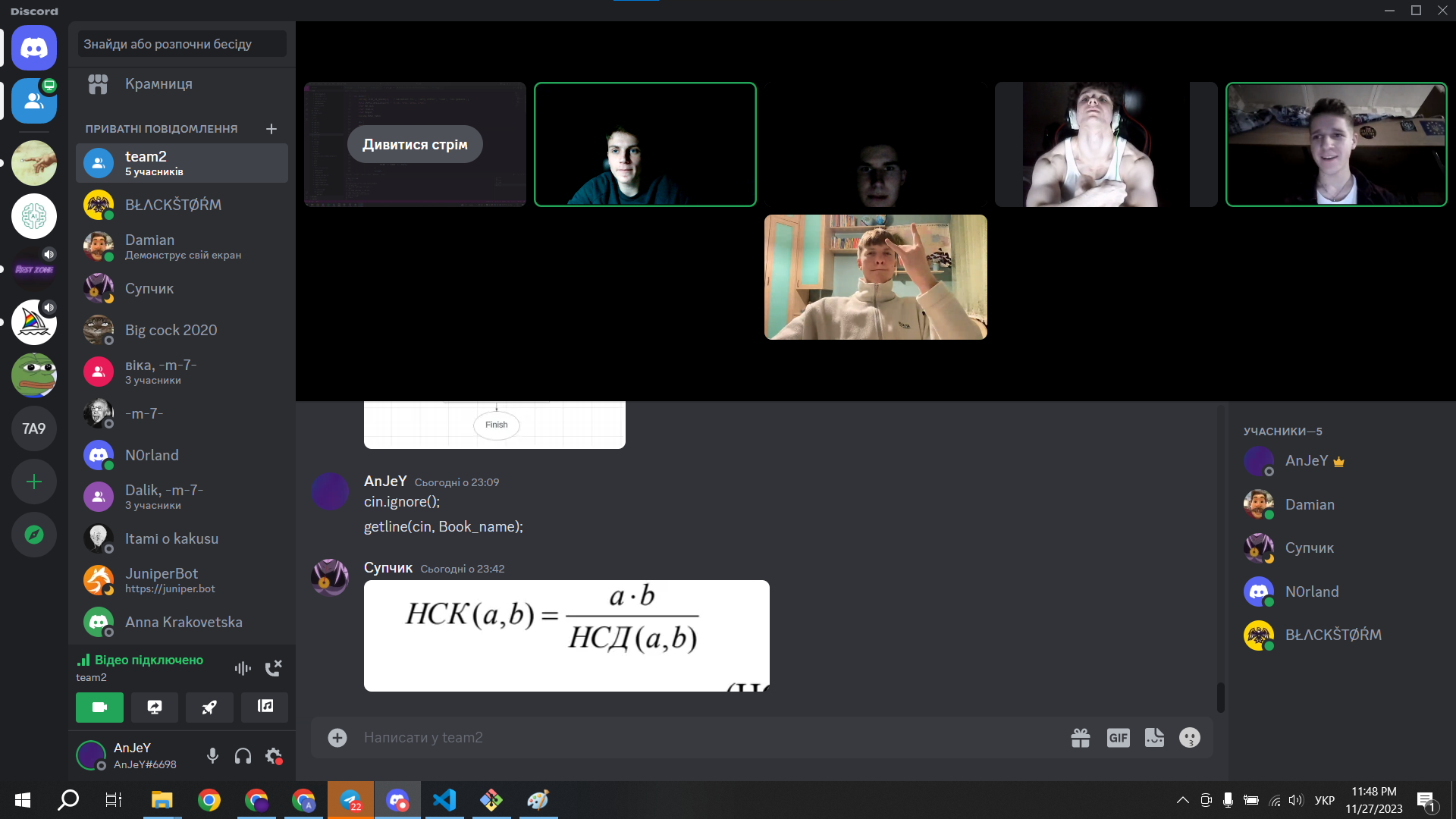


*Image 24. Результати виконання завдання №6 Self Practice Work*

Час затрачений на виконання завдання 15 хв

## **6. Кооперація з командою:**

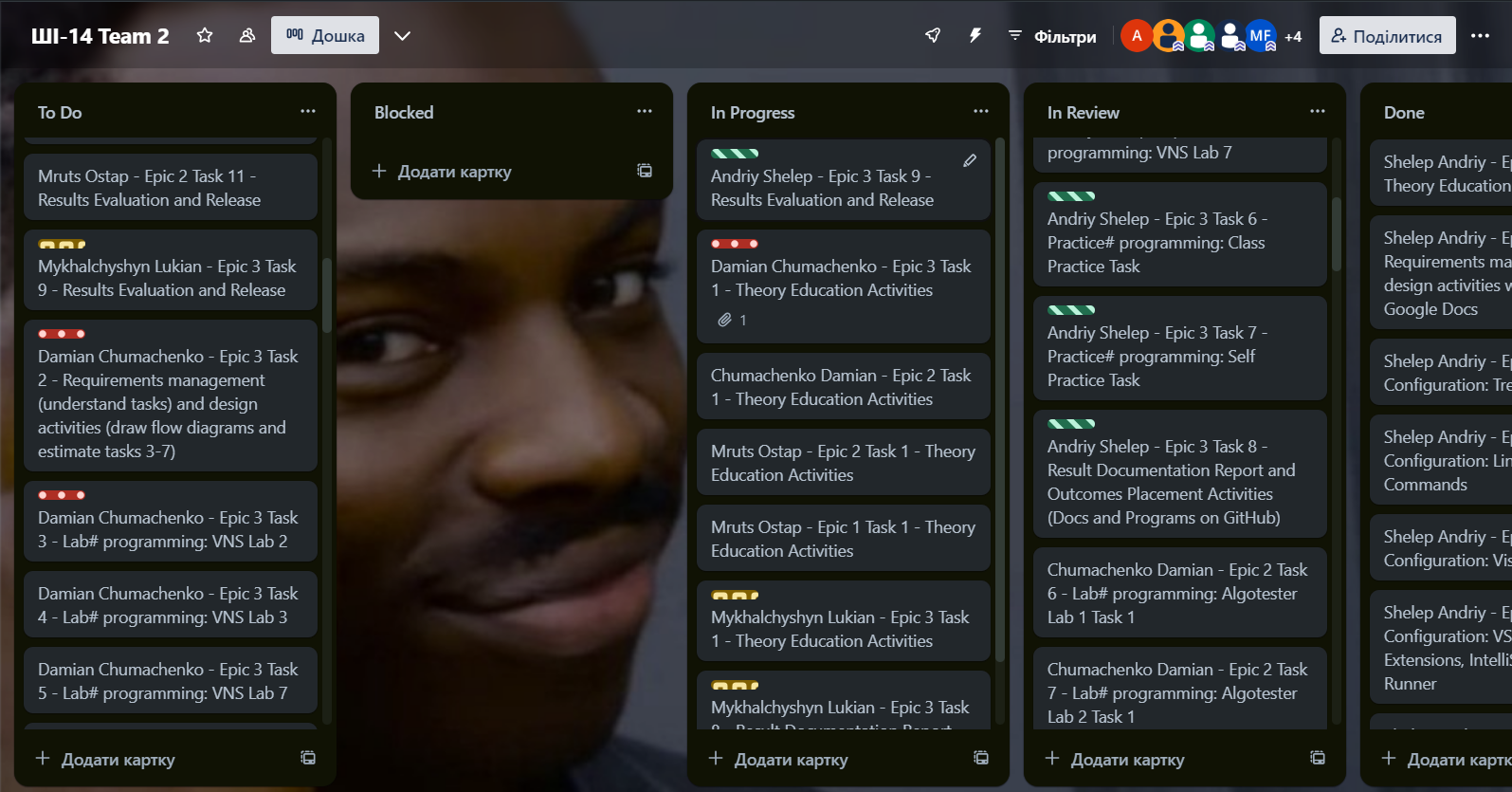
* Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло



*Image 25. Кооперація з командою*

# 

*Image 26. Стан дошки trello*



*Image 27. Стан дошки trello*

# **Висновки:**

Опрацьовано цикли for, while, do while у С++. Ознайомився та використав функції та їхню організацію, перевантаження функції та рекурсію. Виконано лабораторні з ВНС, практичне завдання (менеджер бібліотеки), а також дві задачі з алготестеру з використанням циклів.