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-14

Павлюх Вікторія Ігорівна

Львів 2023

# **Тема роботи:**

Цикли. Вкладені цикли. Оператори continue, break, goto. 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:**

Навчитися використовувати цикли for, while, do while. Ознайомитися та опрацювати функції, 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

- Тема №6: Функція зі змінною кількістю параметрів

2) Індивідуальний план опрацювання теорії:

- Тема №1:Цикли. Вкладені цикли .

o Джерела Інформації

▪ Стаття.

https://studfile.net/preview/5252553/page:3/

o Що опрацьовано:

▪ Цикли у С++: for, while, do while. Різницю між цими циклами. Вкладені цикли

o Статус: Ознайомлена

o Витрачений час: тиждень

- Тема №2: Оператори continue, break, goto.

o Джерела Інформації:

▪ Стаття.

https://schoolboyprog10.blogspot.com/p/36-break-continue-return-systemexit.html

o Що опрацьовано:

▪ Оператори переходу break, goto, continue, return та як вони застосовуються в циклах

o Статус: Ознайомлена

o Витрачений час: 3 дні

- Тема №3:Функція.

o Джерела Інформації:

▪ Відео.

https://youtu.be/G8P6SvdqU9s?feature=shared

▪ Стаття.

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

o Що опрацьовано:

▪ Функції С++, типи повернення функцій, прототипи і визначення функції

o Статус: Ознайомлена

o Витрачений час: тиждень

- Тема №4 Перевантаження функції.

o Джерела Інформації:

▪ Відео.

https://youtu.be/3KJfisev6SI?feature=shared

o Що опрацьовано:

▪ Перевантаження функції

o Статус: Ознайомлена

o Витрачений час: 2 дні

- Тема №5 Рекурсія.

o Джерела Інформації:

▪ Відео.

<https://youtu.be/V7q9w_s0nns?feature=shared>

o Що опрацьовано:

▪ Що таке рекурсія. Рекурсивні функції

o Статус: Ознайомлена

o Витрачений час: 2

# **Виконання роботи:**

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

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

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

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

**Завдання №1 VNS Lab 2 - Task 1**

- Варіант 21

- Деталі завдання: Використовуючи оператор циклу, знайти суму елементів, зазначених у конкретному варіанті. Результат надрукувати, надавши відповідний заголовком.

- Знайти суму 13 членів ряду

- Блок-схема:

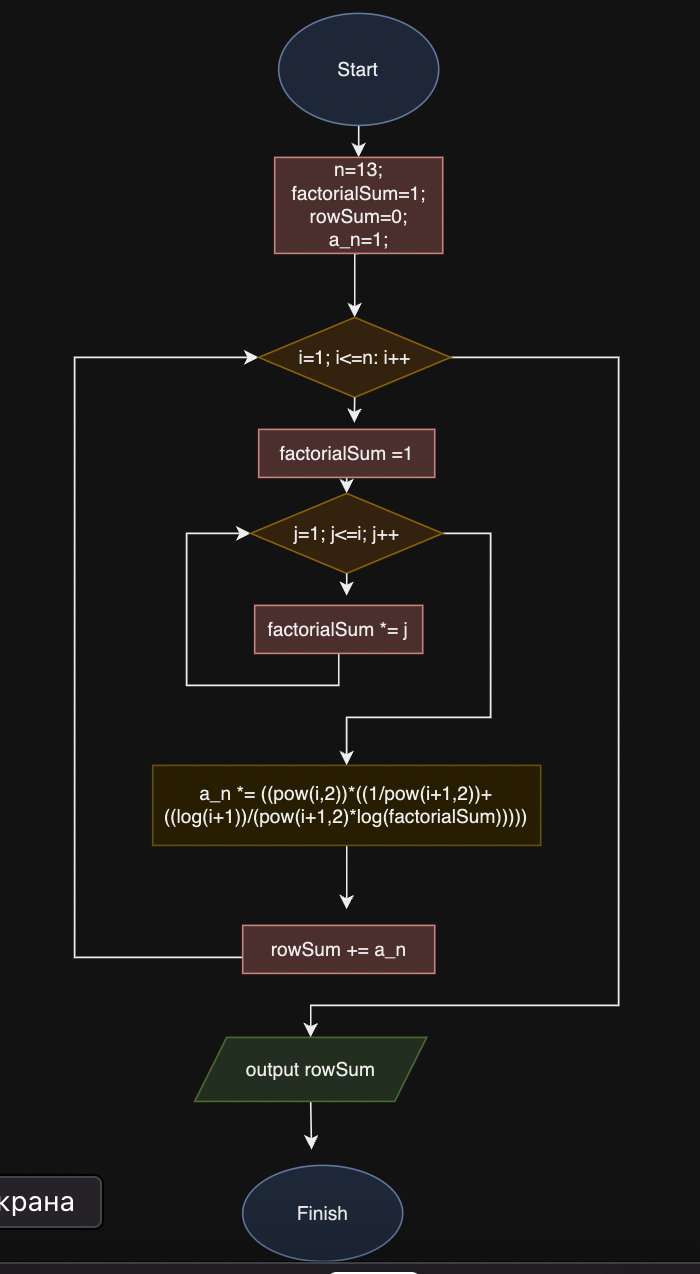


Рис.1. Блок-схема до VNS Lab 2 - Task 1

* Код програми:

*#include* <iostream>

*#include* <cmath>

using namespace std;

int main(){

int n = 13, factorialSum = 1;

double rowSum = 0.0, a\_n = 1.0;

for(int i = 1; i <= n; i++){

factorialSum = 1;

for (int j = 1; j <= n; j++) {

factorialSum \*= j;

}

a\_n \*= ( (pow(i, 2)) \*( (1/pow(i + 1, 2)) + ( (log(i+1))/(pow(i + 1, 2) \*log(factorialSum)) ) ) );

rowSum += a\_n;

}

cout << "Sum of row: " << rowSum << endl;

}

* Результати:

****

Рис.2. Результат тестування VNS Lab 2 - Task 1.

Програма обчислила суму ряду перших 13-и елементів.

**Завдання №2 VNS Lab 3 - Task 1**

- Варіант 21

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

а) для заданого n = 40;

б) для заданої точності ε (ε=0.0001).

- Блок-схема:

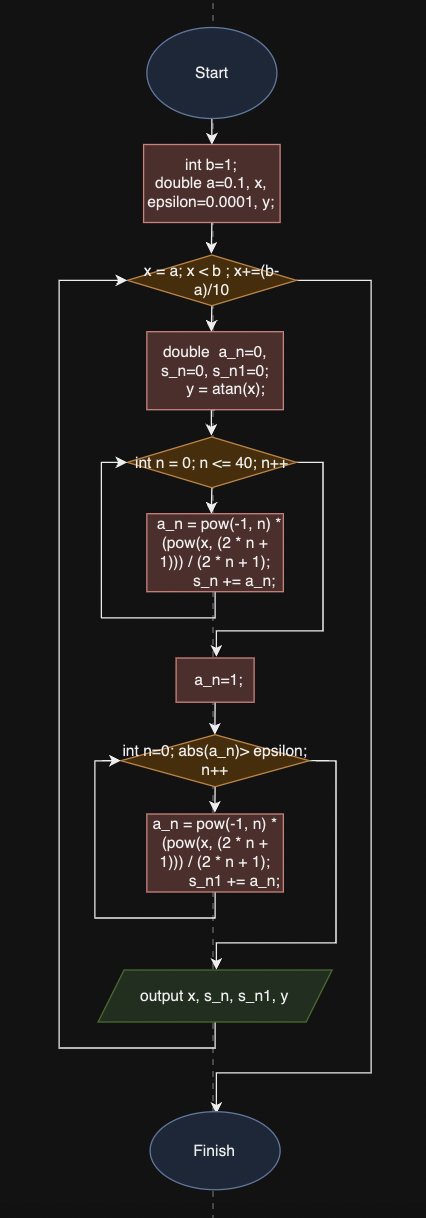


Рис.3. Блок-схема VNS Lab 3 - Task 1

* Код програми:

*#include* <iostream>

*#include* <cmath>

using namespace std;

int main() {

int b=1;

double a=0.1, x, epsilon=0.0001, y;

for ( x = a; x < b ; x+=(b-a)/10) {

double a\_n=0, s\_n=0, s\_n1=0;

y = atan(x);

*//для заданого значення n:*

for (int n = 0; n <= 40; n++) {

a\_n = pow(-1, n) \* (pow(x, (2 \* n + 1))) / (2 \* n + 1);

s\_n += a\_n;

}

*// для заданої точності epsilon:*

a\_n=1;

for(int n=0; abs(a\_n)> epsilon; n++){

a\_n = pow(-1, n) \* (pow(x, (2 \* n + 1))) / (2 \* n + 1);

s\_n1 += a\_n;

}

cout << "x=" << x << " s\_n=" << s\_n << " s\_n1=" << s\_n1 << " y=" << y<<endl;

}

return 0;

}

**-** Результати:

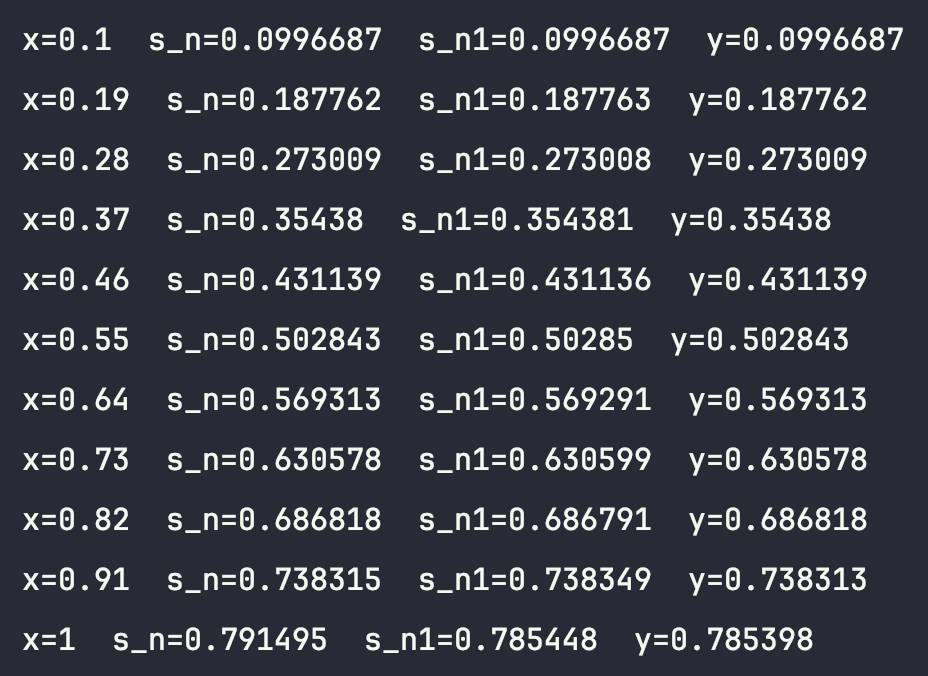
****

Рис.4. Результат тестування VNS Lab 3 - Task 1

**Завдання №3 VNS Lab 7 - Task 1**

- Варіант 2

- Деталі завдання: Розв’язати зазначене у варіанті завдання, використовуючи функції зі змінною кількістю параметрів.

- Блок-схема:

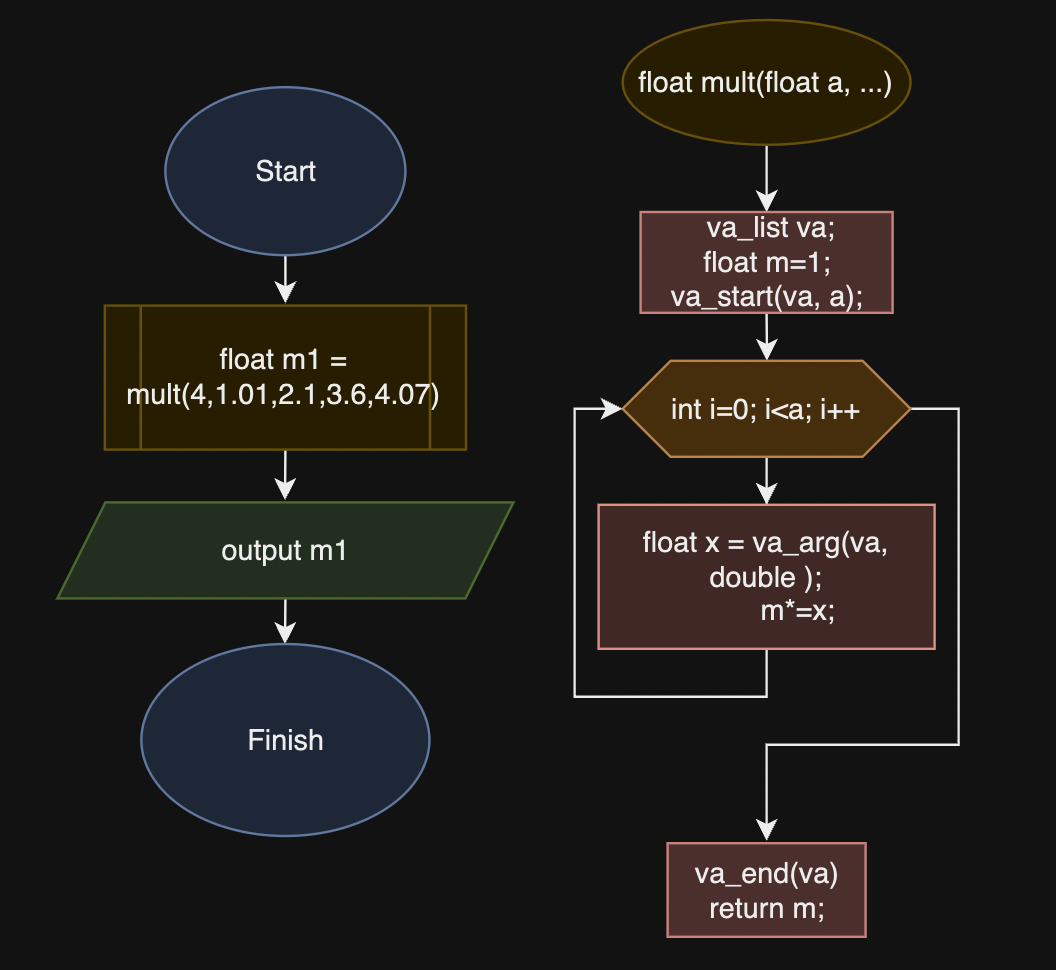


Рис.5. Блок-схема до VNS Lab 7 - Task 1

- Код:

*#include* <iostream>

*#include* <cstdarg>

using namespace std;

float mult(float *a*, ...){

*va\_list* va;

**va\_start**(va, *a*);

float m=1;

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

float x = **va\_arg**(va, double );

m\*=x;

}

**va\_end**(va);

return m;

}

int main() {

float m1 = mult(3,1.01,2.1,3.6);

cout<<m1<<endl;

float m2 = mult(7,1.01,2.1,3.6,4.1,2.5,1.1,3.004);

cout<<m2<<endl;

float m3 = mult(11,1.01,2.1,3.6,4.09,5.12,4.54,7.36,4.1,2.2,1.2,1.1);

cout<<m3<<endl;

return 0;

}

- Результати:



Рис.6. Результат тестування програми.

**Завдання №4 Class Practice Work**

- Без варіантів

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

- Використати наступні оператори: while, do, while, for, for each,

goto.

- Блок-схема:

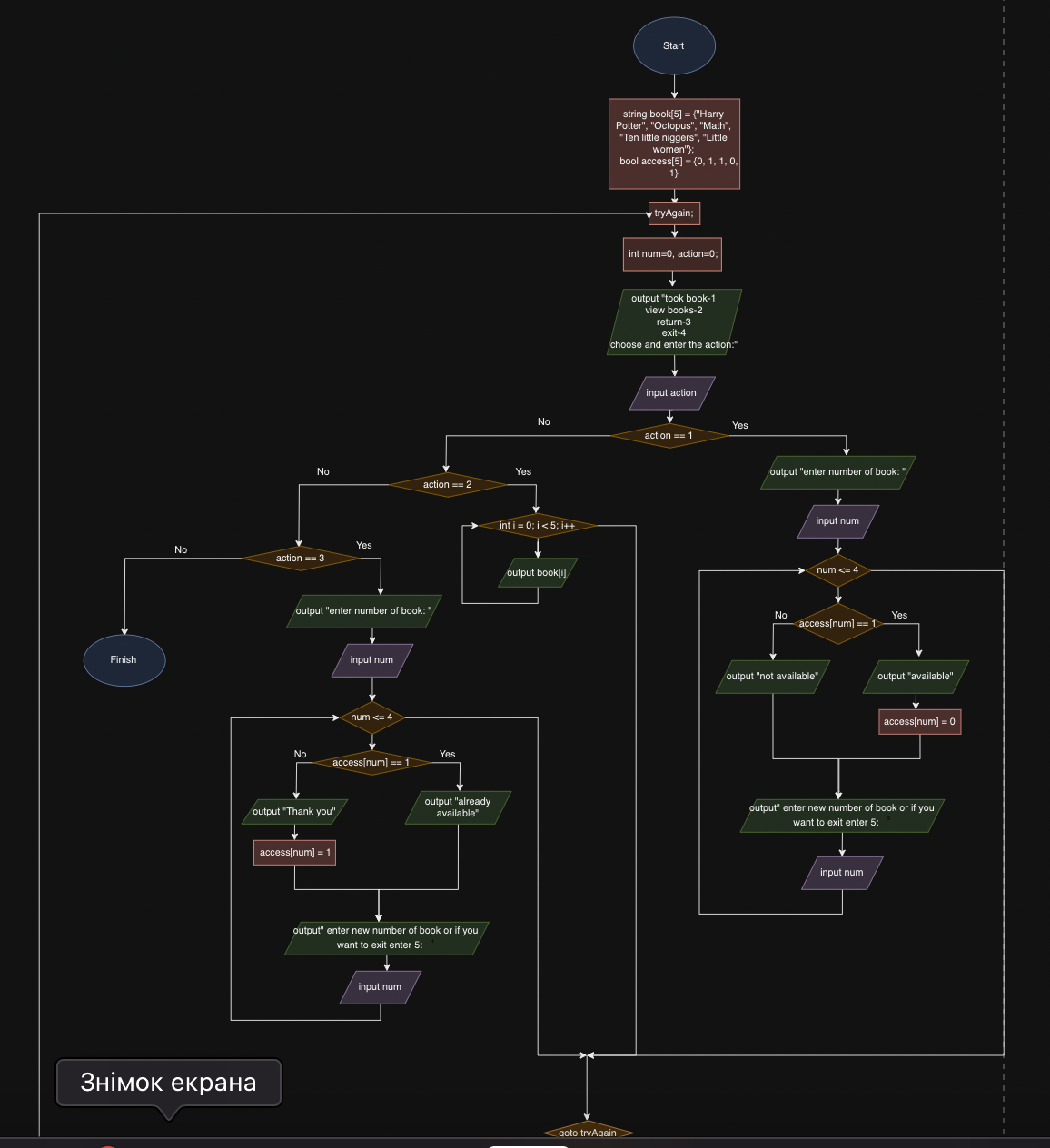


Рис.7. Блок-схема до Class Practice Work

- Код:

*#include* <iostream>

*#include* <string>

using namespace std;

int main() {

*string* book[5] = {"Harry Potter", "Octopus", "Math", "Ten little niggers", "Little women"};

bool access[5] = {0, 1, 1, 0, 1};

int num, action;

tryAgain:

num = 0;

action = 0;

cout << "took book-1" << endl;

cout << "view books-2" << endl;

cout << "return-3" << endl;

cout << "exit-4" << endl;

cout << "choose and enter the action:" << endl;

cin >> action;

if (action == 1) {

cout << "enter number of book: " << endl;

cin >> num;

while (num <= 4) {

if (access[num] == 1) {

cout << "available" << endl;

access[num] = 0;

} else {

cout << "not available" << endl;

}

cout << "enter new number of book or if you want to exit enter 5: " << endl;

cin >> num;

}

} else if (action == 2) {

for (int i = 0; i < 5; i++) {

cout << book[i] << "-" << i << ", ";

}

} else if (action == 3) {

cout << "enter number of book: " << endl;

cin >> num;

while (num <= 4) {

if (access[num] == 1) {

cout << "already available" << endl;

} else {

cout<<"thank you!"<<endl;

access[num] = 1;

}

cout << "enter new number of book or if you want to exit enter 5: " << endl;

cin >> num;

}

}else{

return 0;

}

goto tryAgain;

}

- Результати:

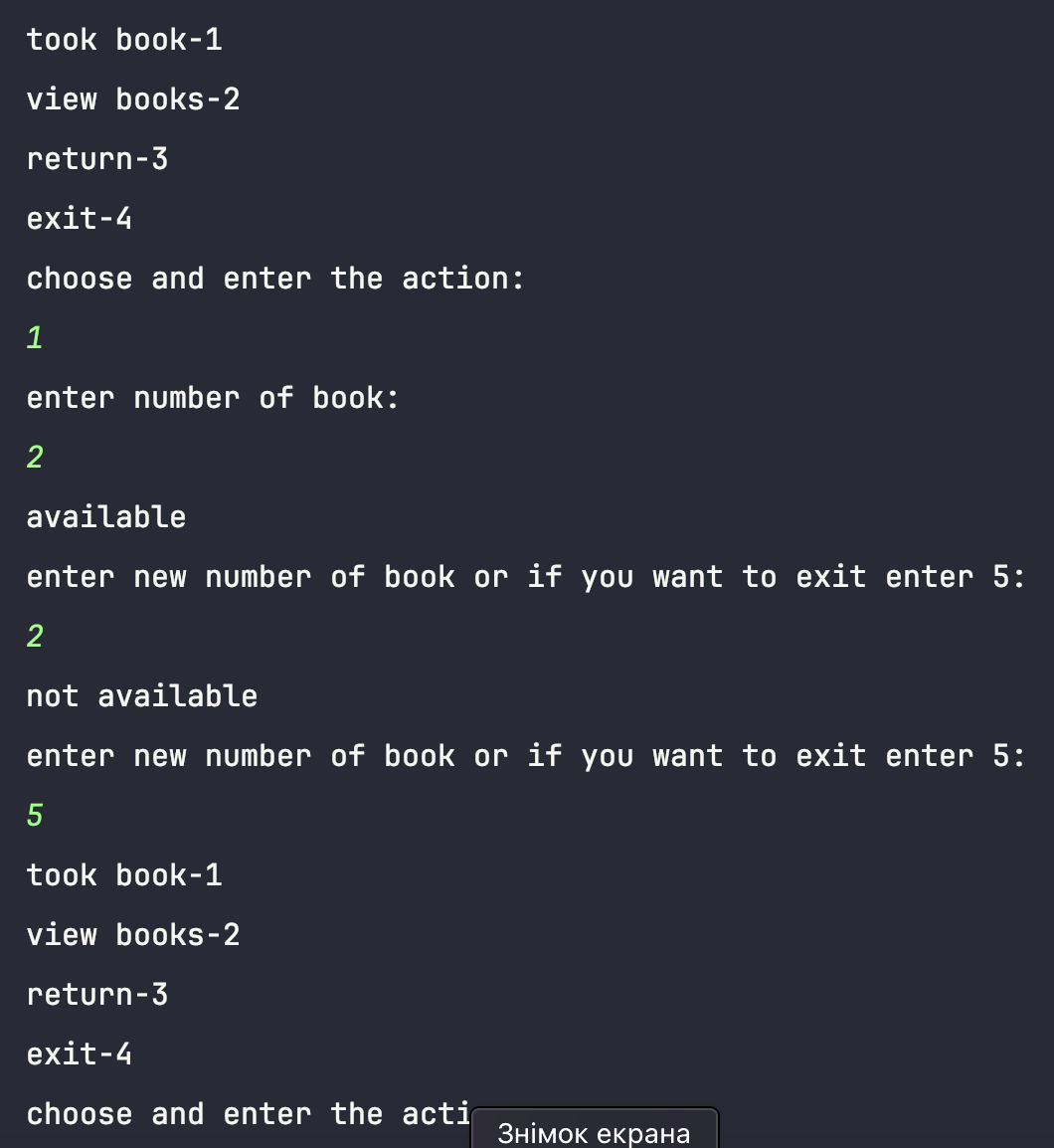


Рис.8. Результати тестування програми.

**Завдання №5 Self Practice Work**

- Задача з алготестеру 0011 - Марічка і печиво.

- Деталі завдання: Марічці не терпиться дізнатися, скільки ж печива вона зможе з’їсти так, щоб Зеник не помітив. Зеник помітить пропажу печива з деякої пачки тоді і тільки тоді, коли Марічка повністю спустошить її.

- Блок-схема:

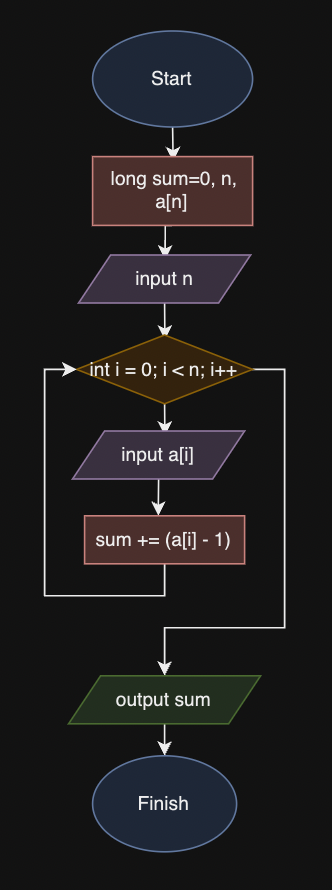


Рис.9. Блок-схема до Self Practice Work.

- Код:

*#include* <iostream>

using namespace std;

int main() {

long sum=0;

long n;

cin >>n;

long a[n];

for (int i = 0; i < n; i++) {

cin >> a[i];

sum += (a[i] - 1);

*// sum += a[i];*

}

cout<<sum;

*// cout<<(sum-n);*

return 0;

}

- Результати:



Рис.10. Результат тестування Self Practice Work.

# **Висновки:**

Навчилася використовувати цикли for, while, do while у С++. Ознайомилася з функціями(звичайними та зі змінною кількістю параметрів), перевантаженням функцій та рекурсією. Виконала лабораторні з ВНС, практичне завдання (менеджер бібліотеки), а також задачу з алготестеру з використанням циклів.