Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

NO_CAPTION null

**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-11

Труш Соломія Володимирівна

# **Тема роботи:**

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

# **Мета роботи:**

Розуміти як працюють і навчитись використовувати цикли, вкладені цикли, функції і їх перевантаження та рекурсію.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

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

1. Індивідуальний план опрацювання теорії:

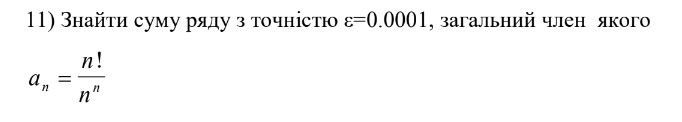
* Тема №1: **Цикли. Вкладені цикли**
  + Джерела Інформації
    - Відео. <https://youtu.be/vLnPwxZdW4Y?si=oXBqeqfnyjZ4cP-G>
    - Стаття 1. <https://www.programiz.com/cpp-programming/for-loop>
    - Стаття 2. <https://www.programiz.com/cpp-programming/do-while-loop>
    - Стаття 3. <https://www.programiz.com/cpp-programming/goto>
    - Лекційний матеріал з ВНС
  + Що опрацьовано:
    - Розумію принципи роботи циклів **for**, **for each, while**, **do-while**, **goto**
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11
  + Звершення опрацювання теми: 30.11
* Тема №2: **Функції. Перевантаження функцій. Рекурсія**
  + Джерела Інформації:
    - Відео 1. <https://youtu.be/vLnPwxZdW4Y?si=oXBqeqfnyjZ4cP-G>
    - Стаття 1. <https://www.programiz.com/cpp-programming/function>
    - Стаття 2. <https://acode.com.ua/urok-108-perevantazhennya-funktsij/>
    - Стаття 3. <https://www.programiz.com/cpp-programming/recursion>
    - Лекційний матеріал з ВНС
    - [chat.openai.com](https://chat.openai.com/)
    - [bard.google.com](https://bard.google.com/?hl=uk)
  + Що опрацьовано:
    - Розумію що таке функція та принципи її роботи, що таке перевантаження функції, що таке рекурсія
  + Статус: Ознайомлений
  + Початок опрацювання теми: Дата 25.11
  + Звершення опрацювання теми: Дата 30.11

# **Виконання роботи:**

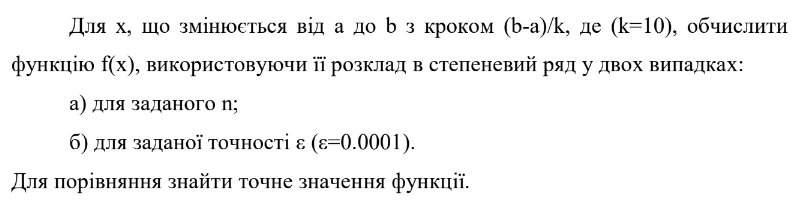
## **1. Опрацювання завдання та вимог до програм та середовища:**

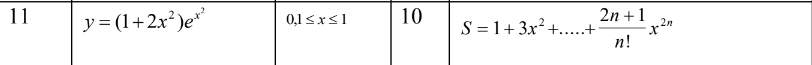
**VNS Lab 2 - Task 1-N**

* Варіант 11
* Деталі завдання



**VNS Lab 3 - Task 1-N**

* Варіант 11
* Деталі завдання
* 



**VNS Lab 7 - Task 1-N**

* Варіант 11
* Деталі завдання

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

кількістю параметрів.

Написати функцію sum зі змінною кількістю параметрів, що знаходить суму

заданих звичайних дробів. Написати викликаючу функцію main, що

звертається до функції sum не менше трьох разів з кількістю параметрів 5, 10,

12.

**VNS Lab 7 - Task 2-N**

* Варіант 11
* Деталі завдання

Написати перевантажені функції й основну програму, що їх викликає.

а) для множення десяткових дробів;

б) для множення звичайних дробів.

**Class Practice Work**

* Варіант завдання
* Деталі завдання

**Менеджмент бібліотеки**

*Задача*

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.

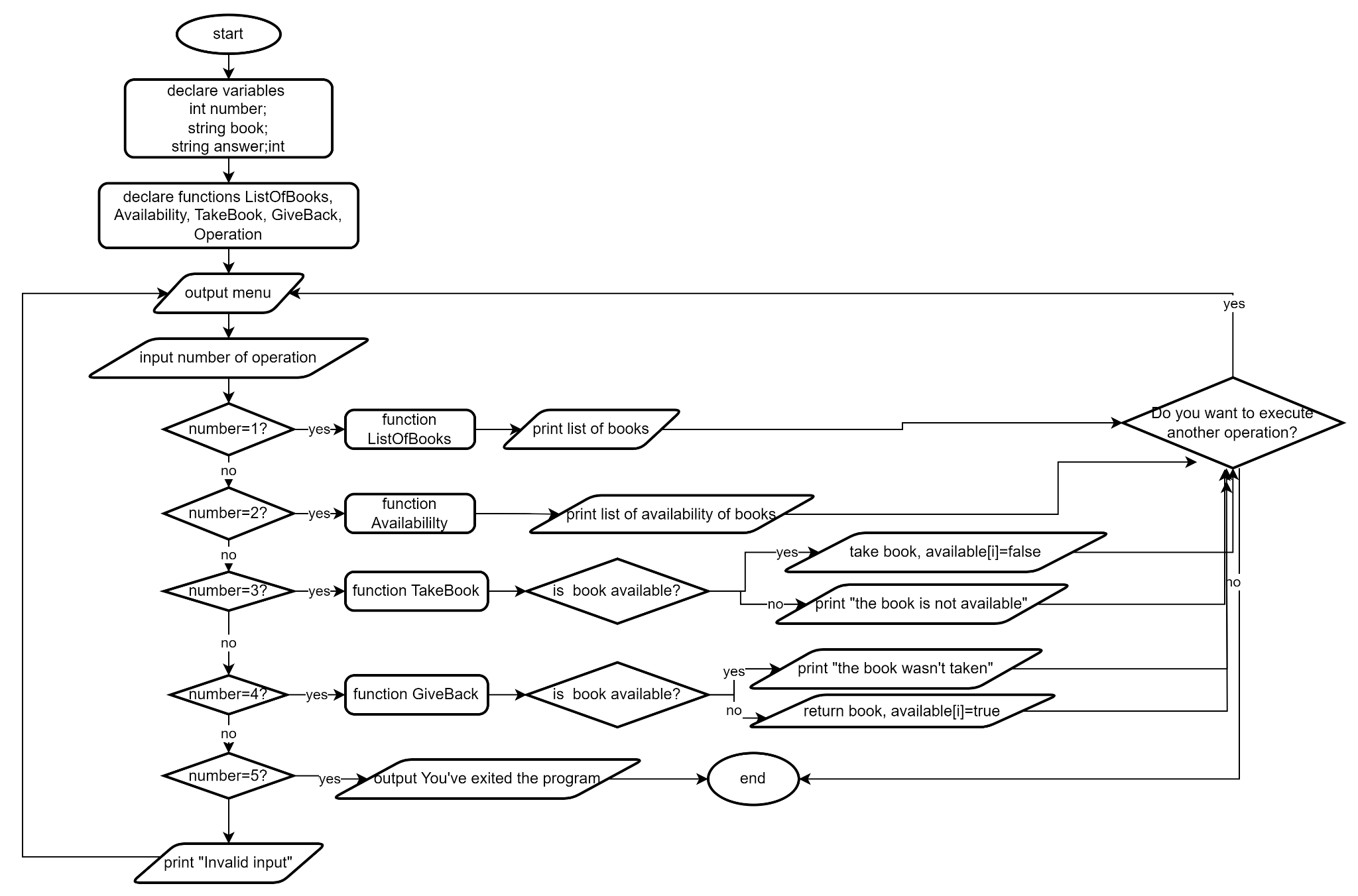
**Self Practice Work**

* Варіант завдання
* Деталі завдання
* Важливі деталі для врахування в імплементації програми

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

**Class Practice Work**

* Блок-схема

****

*Figure 1. Flowchart до Class Practice Work*

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

Середовище сконфігуровано під час виконання Epic 1.

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

**VNS Lab 2 - Task 1-N**

#include <iostream>

#include <cmath>

using namespace std;

int main () {

unsigned int n=1;

double factorial=1;

double an;

double sum=0;

cout << "Enter n: ";

cin >> n;

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

factorial\*=i;

an = factorial/pow(i,i);

sum += an;

}

cout << sum;

return 0;

}

**VNS Lab 3 - Task 1-N**

#include <iostream>

#include <cmath>

using namespace std;

double Factorial(int i) {

double f=1;

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

f\*=j;

}

return f;

}

int main () {

double x, sn=0, se=0, y=0, a=0.1, b=1, eps=0.0001;

int n=10, k=10;

double e = exp(1.0) ;

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

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

sn += ((2\*i+1)/(Factorial(i))\*pow(x,2\*i));

}

int l=0;

double t=1;

while (fabs(t)>= eps) {

se += t;

t -= ((2\*l+1)/(Factorial(l))\*pow(x,2\*l));

l++;

}

y = (1+2\*pow(x,2))\*pow(e,pow(x,2));

cout << "X = " << x << " SN = " << sn << " SE = " << se << " Y = " << y << endl;

}

return 0;

}

**VNS Lab 7 - Task 1-N**

#include <iostream>

#include <cstdarg>

using namespace std;

double sum (int k, ...) {

va\_list elements;

va\_start(elements,k);

double ssum = 0;

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

ssum += va\_arg(elements, double);

}

va\_end(elements);

return ssum;

}

int main () {

cout << "Sum of 1.0/2, 2.0/4, 3.0/6, 4.0/8, 5.0/10 = " << sum (5, 1.0/2, 2.0/4, 3.0/6, 4.0/8, 5.0/10) << endl;

cout << "Sum of 1.0/2, 2.0/4, 3.0/6, 4.0/8, 5.0/10, 6.0/12, 7.0/14, 8.0/16, 9.0/18, 10.0/20 = " << sum (10, 1.0/2, 2.0/4, 3.0/6, 4.0/8, 5.0/10, 6.0/12, 7.0/14, 8.0/16, 9.0/18, 10.0/20) << endl;

cout << "Sum of 1.0/2, 2.0/4, 3.0/6, 4.0/8, 5.0/10, 6.0/12, 7.0/14, 8.0/16, 9.0/18, 10.0/20, 11.0/22, 12.0/24 = " << sum (12, 1.0/2, 2.0/4, 3.0/6, 4.0/8, 5.0/10, 6.0/12, 7.0/14, 8.0/16, 9.0/18, 10.0/20, 11.0/22, 12.0/24) << endl;

return 0;

}

**VNS Lab 7 - Task 2-N**

#include <iostream>

#include <cmath>

#include <complex>

using namespace std;

int addition (int a, int b) {

int sum = a+b;

return sum;

}

complex<double> addition (complex<double> a , complex<double> b ) {

complex<double> sum = a+b;

return sum;

}

int main () {

complex<double> complex\_number=addition ({2.0, 3.0}, {5.0, 8.0});

cout << "Addition of integers : " << addition (4,5) << endl;

cout << "Addition of complex numbers : " << complex\_number.real() << "+" << complex\_number.imag() << "i" << endl;

}

}

**Class Practice Work**

**#include <iostream>**

**#include <vector>**

**#include <string>**

**using namespace std;**

**int number;**

**string book;**

**string answer;**

**vector<string> books = {"Harry Potter", "The Hobbit", "Pride and Prejudice", "The Lord of Rings", "The Adventures of Sherlock Holmes", "Hansel and Gretel", "The Fight Club"};**

**vector<bool> availability = {true, true, true, true, true, true, true };**

**void ListOfBooks () {**

**for(string k : books) {**

**cout<<k<<" " << endl;**

**}**

**}**

**void Availability () {**

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

**cout << "\n" << books[i] << " - " ;**

**if (availability[i]==true) {**

**cout << "available";**

**}**

**else {**

**cout << "not available";**

**}**

**}**

**cout << "\n";**

**}**

**void TakeBook (int i, string book) {**

**if (availability[i]==false) {**

**cout << "The book is not available" << endl;**

**return;**

**}**

**else {**

**cout << "You took the book "<< book << endl;**

**availability[i] = false;**

**return;**

**}**

**}**

**void GiveBack (int i, string book) {**

**if (availability[i]==true) {**

**cout << "The book wasn't taken" << endl;**

**return;**

**}**

**else {**

**cout << "You returned the book "<< book << endl;**

**availability[i] = true;**

**return;**

**}**

**}**

**void Operation (int number) {**

**switch (number) {**

**case 1 :**

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

**ListOfBooks();**

**break;**

**case 2 :**

**Availability();**

**break;**

**case 3 :**

**cout << "Enter the name of the book : " << endl;**

**getline(cin, book);**

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

**if (books[i]==book) {**

**TakeBook(i, book);**

**break;**

**}**

**}**

**break;**

**case 4 :**

**cout << "Enter the name of the book : " << endl;**

**getline(cin, book);**

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

**if (book==books[l]) {**

**GiveBack(l, book);**

**return;**

**}**

**}**

**break;**

**}**

**}**

**int main () {**

**while (true) {**

**menu :**

**cout << "\n";**

**cout << "MENU" << endl;**

**cout << "1 List of books" << endl;**

**cout << "2 Availability of books" << endl;**

**cout << "3 Take the book" << endl;**

**cout << "4 Return the book " << endl;**

**cout << "5 Exit" << "\n" << endl;**

**cout << "Enter number of the operation : " << endl;**

**cin >> number;**

**cin.ignore();**

**if (number!=5) {**

**Operation(number);**

**}**

**else if (number==5) {**

**cout << "You've exited the program" << endl;**

**return 0;**

**}**

**else if (number!=1 && number!=2 && number!=3 && number!=4) {**

**cout << "Invalid input. Try again" << endl;**

**goto menu;**

**}**

**do {**

**cout << "\n";**

**cout << "Do you want to execute another operation? (y/n)" << endl;**

**cin >> answer;**

**if (answer[0]=='y') {**

**break;**

**}**

**else if (answer[0]=='n') {**

**cout << "You've exited the program";**

**return 0;**

**}**

**else {**

**cout << "Invalid input" << endl << "\n";**

**goto menu;**

**}**

**} while (answer[0]=='y');**

**}**

**}**

**Self Practice Work**

**#include <iostream>**

**using namespace std;**

**int main() {**

**long pachky=0, shavaty=0;**

**cin >> pachky;**

**long pechenky[pachky];**

**for (long i=0; i<pachky; i++) {**

**cin >> pechenky [i];**

**}**

**for (long j=0; j<pachky; j++) {**

**shavaty += pechenky[j];**

**}**

**shavaty -= pachky;**

**cout << shavaty;**

**}**

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

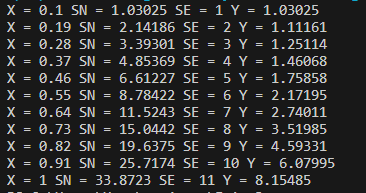
**VNS Lab 2 - Task 1-N**



*Figure 2. Результат VNS Lab 2 - Task 1-N*

Планований час виконання - 2 години, фактичний - 2 год

**VNS Lab 3 - Task 1-N**

****

*Figure 3. Результат VNS Lab 3 - Task 1-N*

Планований час виконання - 3 години, фактичний - 4 години

**VNS Lab 7 - Task 1-N**

****

*Figure 4. Результат VNS Lab 7 - Task 1-N*

Планований час виконання - 3 години, фактичний - 3 години

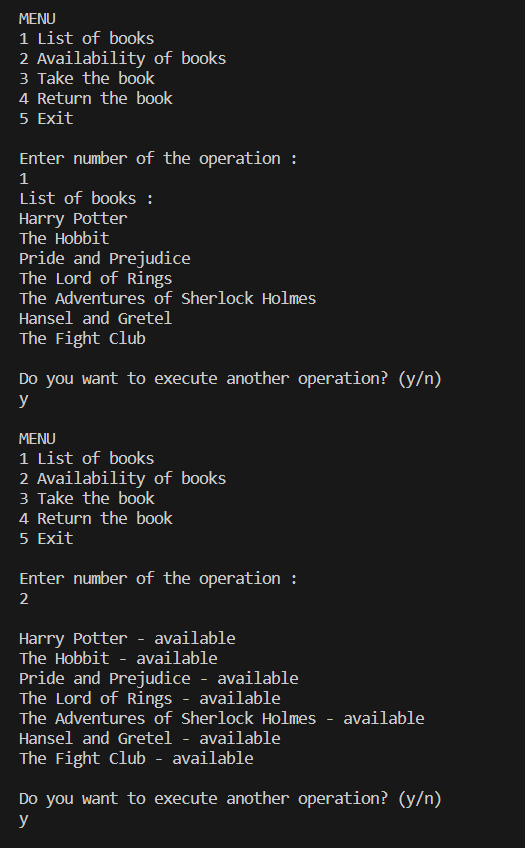
**VNS Lab 7 - Task 2-N**

****

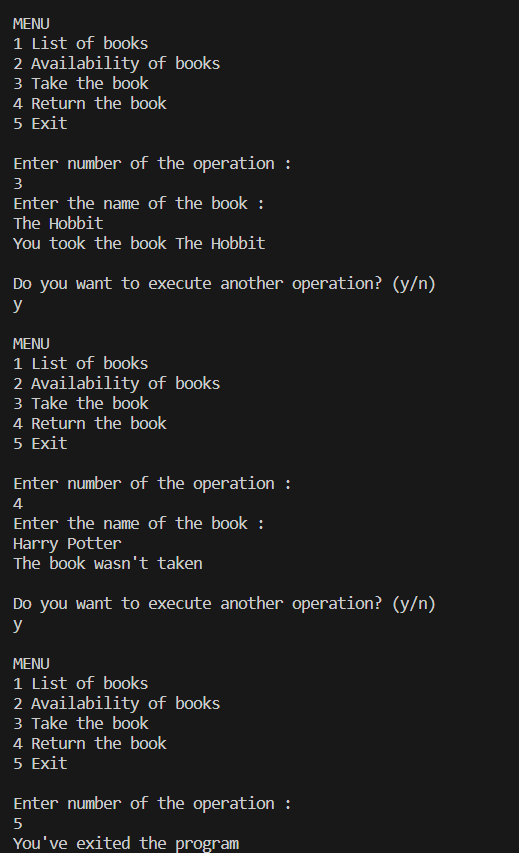
*Figure 5. Результат VNS Lab 7 - Task 2-N*

Планований час виконання - 1 година, фактичний - 1 година

**Class Practice Work**

****

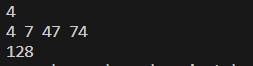
*Figure 6. Результат Class Practice Work*

****

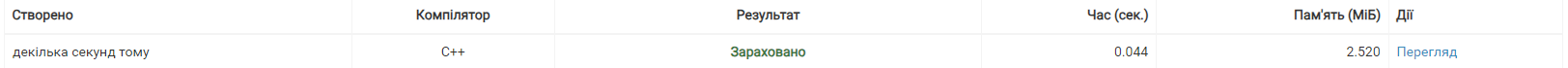
*Figure 7. Результат Class Practice Work*

Планований час виконання - 4 годин, фактичний - 6 годин

**Self Practice Work**

****

*Figure 8. Output Self Practice Work*



*Figure 9. Algotester result Self Practice Work*

Планований час виконання завдання - 15 хв, фактичний час виконання - 15 хв

# **Висновки:**

Ознаймоилась з циклами, вкладеними циклами, функціями, перевантаженими функціями, рекурсією,va\_list, vector, goto. Використала свої знання під час виконання лабораторних робіт.