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:*** «Epic 3. Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія. Змінні»

***Виконав:***

студент групи ШІ-14

Михальчишин Лук’ян Павлович

**Тема роботи:**

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

**Мета роботи:**

Навчитись використовувати прості цикли та вкладені цикли, застосовувати перевантаження функції і рекурсію, вміти створювати функції

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Theory Education Activities
* Тема №2:  Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)
* Тема №3: Lab# programming: VNS Lab 2
* Тема №4: Lab# programming: VNS Lab 3
* Тема №5  Lab# programming: VNS Lab 7
* Тема №6 Practice# programming: Class Practice Task
* Тема №7 Practice# programming: Self Practice Task

1. Індивідуальний план опрацювання теорії:

* **Тема №1: Theory Education Activities**

1. Джерела Інформації

Статті

Курс: <https://youtu.be/kK-Xo25DIXc?si=T78UkzsQRIKmp5xO>

<https://youtu.be/G8P6SvdqU9s?si=LiVnFbygKQN3AmNS>

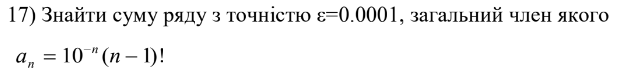
<https://youtu.be/3KJfisev6SI?si=jOkng82H3DEmgdMb>

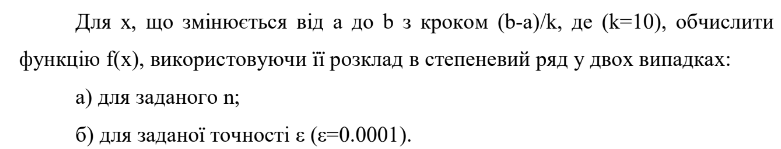
<https://youtu.be/V7q9w_s0nns?si=SJJHiizI55H1u6Bo>

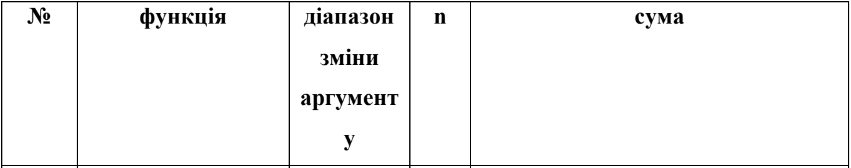
1. Що опрацьовано: Поняття рекурсії, перевантаження функції, застосування Debuger, вкладені цикли
2. Статус: Ознайомлений

**Виконання роботи:**

* 1. **Опрацювання завдання та вимог до програм та середовища:**
* Task 2 - Requirements management and design activities with Draw.io and Google Docs
  + Створити блок-схеми до кодів
* Task 3: Lab# programming: VNS Lab 2

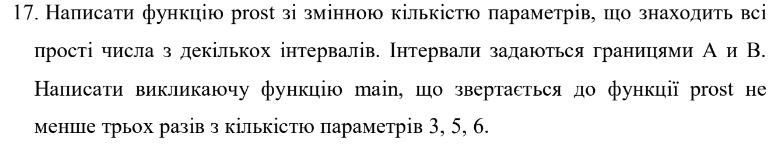


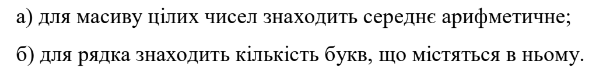
* Task 4: Lab# programming: VNS Lab 3





* Task 5: Lab# programming: VNS Lab 7





* Task 6: Practice# programming: Class Practice Task
  + Написати програму, яка допомагає читачу зробити певні дії над книжками, які він ввів
* Task 7: Practice# programming: Self Practice Task
  + Депутатські гроші

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

Блок-схема

* Task 3  Lab# programming: VNS Lab 2

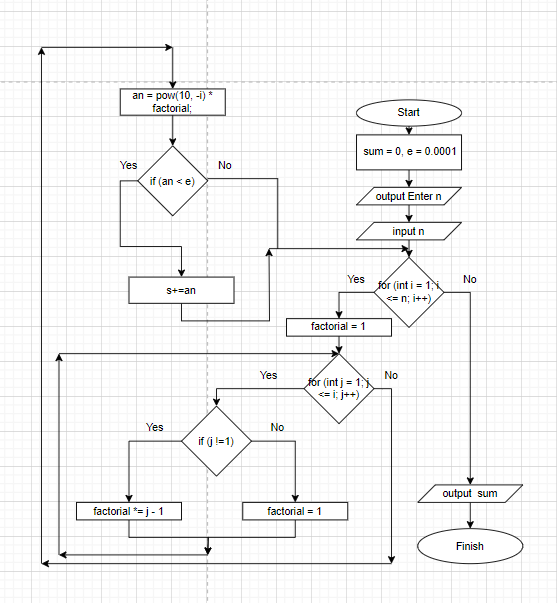


Рисунок 1: Блок схема до програми №1

* Task 4: Lab# programming: VNS Lab 3

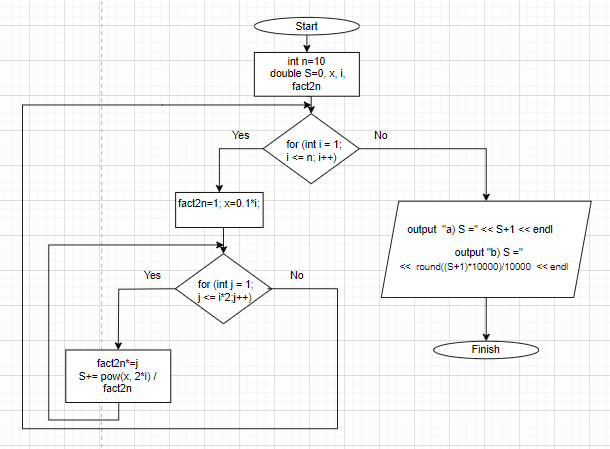


Рисунок 2: Блок схема до програми №2

* Task 5: Lab# programming: VNS Lab 7

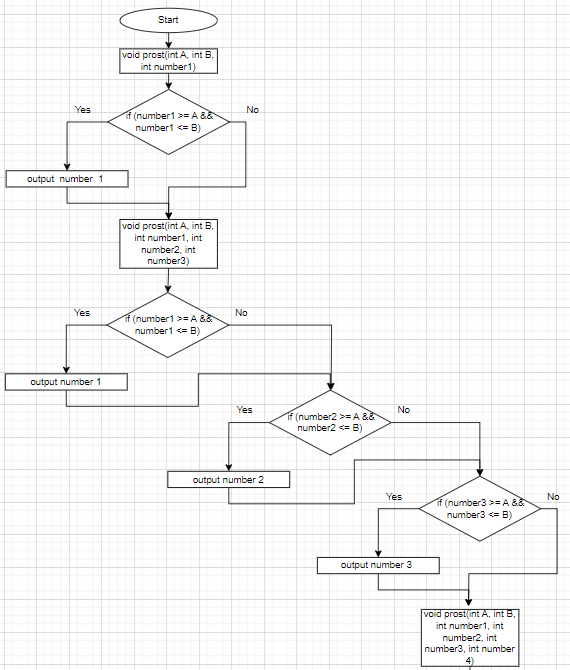


Рисунок 3: Блок схема до програми № 3

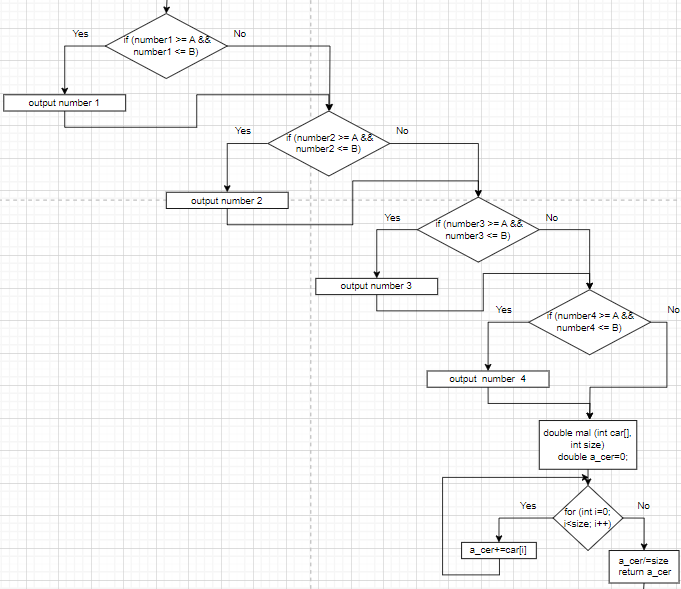


Рисунок 4: Блок схема до програми № 3

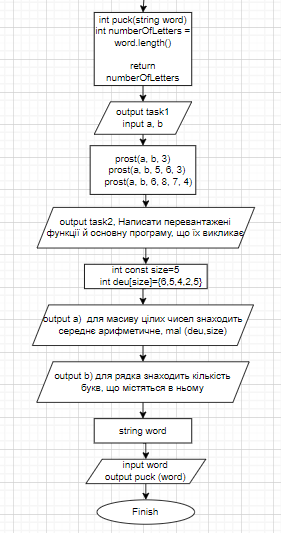
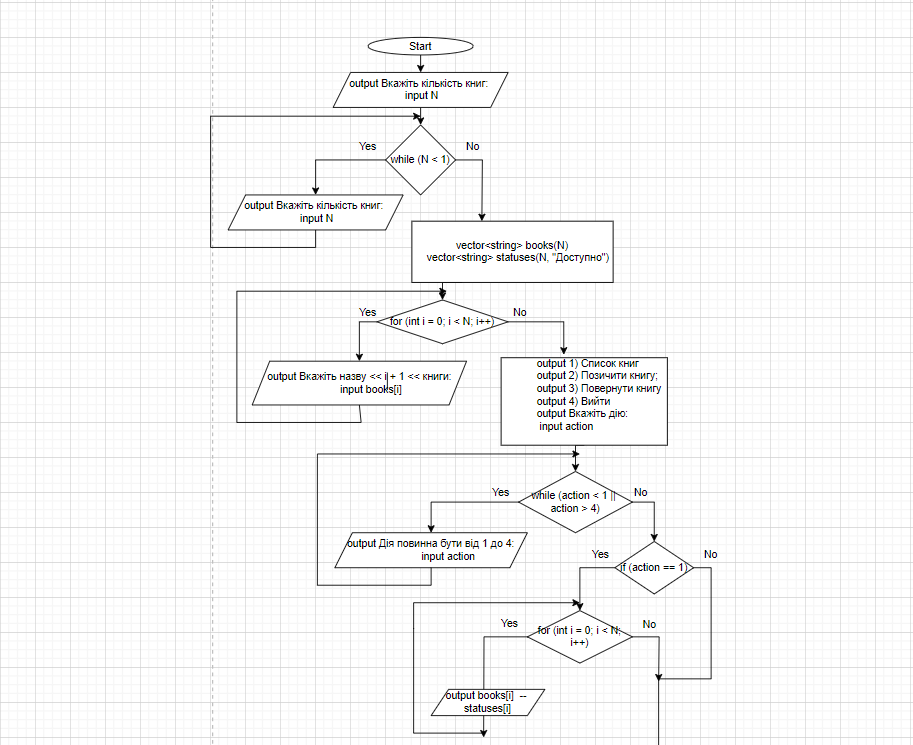


Рисунок 5: Блок схема до програми № 3

* Task 6: Practice# programming: Class Practice Task



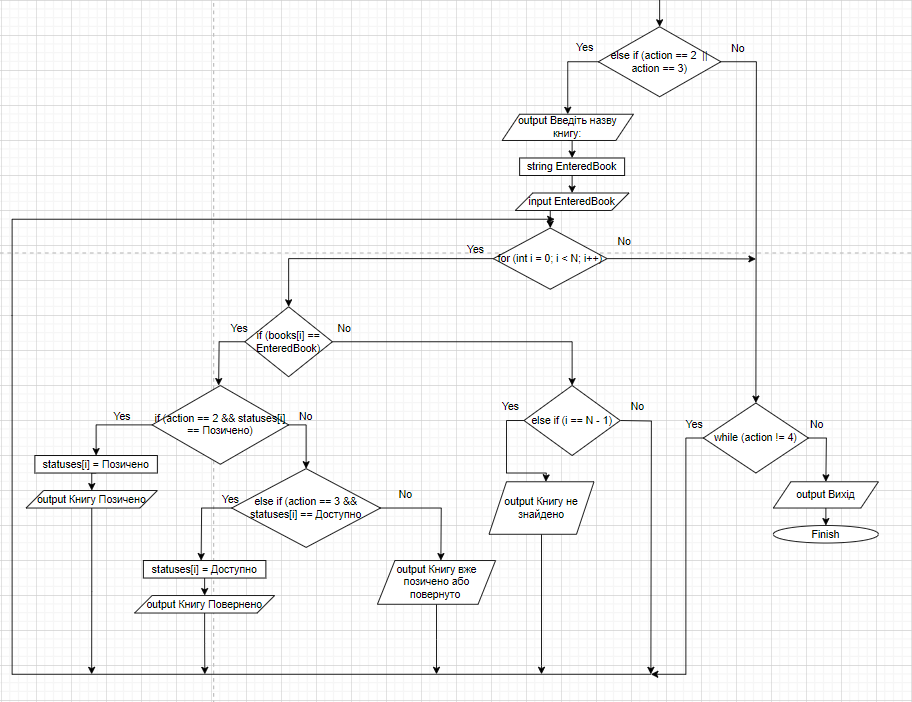


Рисунок 6: Блок схема до програми № 5

* Task 7: Practice# programming: Self Practice Task

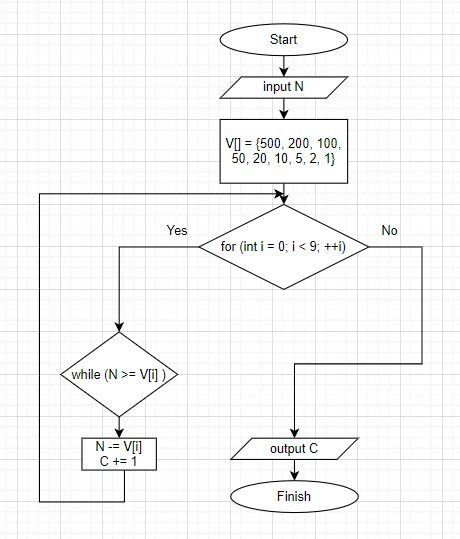


Рисунок 7: Блок схема до програми № 6

* 1. **Код програм з посиланням на зовнішні ресурси:**
* Task 3  Lab# programming: VNS Lab 2
* #include <iostream>
* #include <cmath>
* using namespace std;
* // умова an = 10^-n \* (n-1)!
* int main()
* {
* int n;
* double an, sum = 0, factorial, e = 0.0001;
* cout << "Enter n \n";
* cin >> n;
* for (int i = 1; i <= n; i++)
* {
* factorial = 1;
* for (int j = 1; j <= i; j++)
* {
* if (j != 1)
* {
* factorial \*= j - 1;
* }
* else{ factorial=1;}
* }
* an = pow(10, -i) \* factorial;
* if (an < e)
* {
* sum += an;
* cout << an << " ";
* }
* }
* cout << endl
* << sum << endl;
* return 0;
* }
* Task 4: Lab# programming: VNS Lab 3

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

    int n = 10;

    double S=0, x, i, fact2n;

    for (int i = 1; i <= n; i++)

    {

        fact2n=1;

        x=0.1\*i;

        for (int j = 1; j <= i\*2; j++)

        {

           fact2n\*=j;

        }

        S+= pow(x, 2\*i) / fact2n;

    }

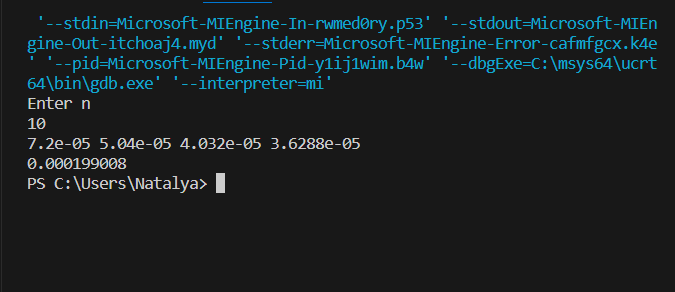
    cout << "a) S =" << S+1 << endl;

    cout << "b) S =" << round((S+1)\*10000)/10000 << endl;

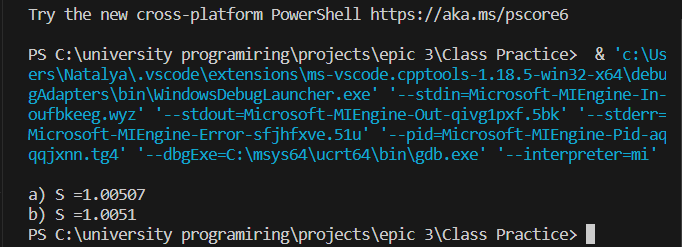
    return 0;

}

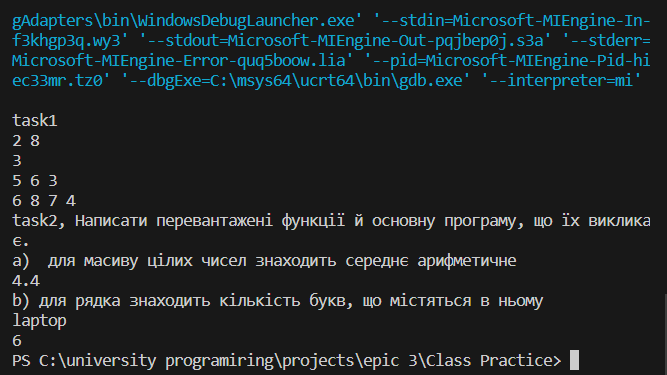
* Task 5: Lab# programming: VNS Lab 7
* #include <iostream>
* using namespace std;
* void prost(int A, int B, int number1)
* {
* if (number1 >= A && number1 <= B)
* cout << number1<< " ";
* }
* void prost(int A, int B, int number1, int number2, int number3)
* {
* if (number1 >= A && number1 <= B)
* cout << number1<< " ";
* if (number2 >= A && number2 <= B)
* cout << number2<< " ";
* if (number3 >= A && number3 <= B)
* cout << number3<< " ";
* }
* void prost(int A, int B, int number1, int number2, int number3, int number4)
* {
* if (number1 >= A && number1 <= B)
* cout << number1 << " ";
* if (number2 >= A && number2 <= B)
* cout << number2 << " ";
* if (number3 >= A && number3 <= B)
* cout << number3 << " ";
* if (number4 >= A && number4 <= B)
* cout << number4 << " ";
* }
* double mal (int car[], int size)
* {
* double a\_cer=0;
* for (int i=0; i<size; i++)
* {
* a\_cer+=car[i];
* }
* a\_cer/=size;
* return a\_cer;
* }
* int puck(string word)
* {
* int numberOfLetters = word.length();
* return numberOfLetters;
* }
* int main()
* {
* cout << "task1\n";
* int a, b;
* cin >> a >> b;
* prost(a, b, 3);
* cout << endl;
* prost(a, b, 5, 6, 3) ;
* cout << endl;
* prost(a, b, 6, 8, 7, 4);
* cout << endl;
* cout << "task2, Написати перевантажені функції й основну програму, що їх викликає.\n";
* int const size=5;
* int deu[size]={6,5,4,2,5};
* cout << "a)  для масиву цілих чисел знаходить середнє арифметичне "<< endl << mal (deu,size) << endl;
* cout << "b) для рядка знаходить кількість букв, що містяться в ньому\n";
* string word;
* cin >> word;
* cout <<  puck(word);
* return 0;
* }
* Task 6: Practice# programming: Class Practice Task
* #include <iostream>
* #include <string>
* #include <vector>
* using namespace std;
* int main() {
* int N, action;
* cout << "Вкажіть кількість книг: ";
* cin >> N;
* while (N < 1) {
* cout << "Кількість книг повинна бути хотяб 1: ";
* cin >> N;
* }
* vector<string> books(N);
* vector<string> statuses(N, "Доступно");
* for (int i = 0; i < N; i++) {
* cout << "Вкажіть назву " << i + 1 << " книги: ";
* cin >> books[i];
* }
* do {
* cout << "\n1) Список книг\n";
* cout << "2) Позичити книгу\n";
* cout << "3) Повернути книгу\n";
* cout << "4) Вийти\n\n";
* cout << "Вкажіть дію: ";
* cin >> action;
* while (action < 1 || action > 4) {
* cout << "Дія повинна бути від 1 до 4: ";
* cin >> action;
* }
* if (action == 1) {
* for (int i = 0; i < N; i++) {
* cout << books[i] << " -- " << statuses[i] << "\n";
* }
* } else if (action == 2 || action == 3) {
* cout << "Введіть назву книгу: ";
* string EnteredBook;
* cin >> EnteredBook;
* for (int i = 0; i < N; i++) {
* if (books[i] == EnteredBook) {
* if (action == 2 && statuses[i] == "Доступно") {
* statuses[i] = "Позичено";
* cout << "Книгу " << books[i] << " Позичено\n";
* } else if (action == 3 && statuses[i] == "Позичено") {
* statuses[i] = "Доступно";
* cout << "Книгу " << books[i] << " Повернено\n";
* } else {
* cout << "Книгу вже позичено або повернуто\n";
* }
* break;
* } else if (i == N - 1) {
* cout << "Книгу не знайдено\n";
* }
* }
* }
* } while (action != 4);
* cout << "Вихід\n";
* return 0;
* }
* Task 7: Practice# programming: Self Practice Task
* #include <iostream>
* using namespace std;
* int main()
* {
* int N, C = 0;
* cin >> N;
* int V[] = {500, 200, 100, 50, 20, 10, 5, 2, 1};
* for (int i = 0; i < 9; ++i)
* {
* while (N >= V[i])
* {
* N -= V[i];
* C += 1;
* }
* }
* cout << C;
* }
  1. **Результати виконання завдань, тестування та фактично затрачений час:**
* Task 3  Lab# programming: VNS Lab 2



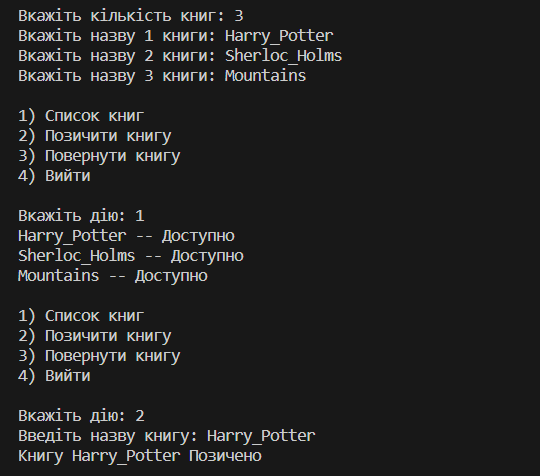
* Task 4: Lab# programming: VNS Lab 3

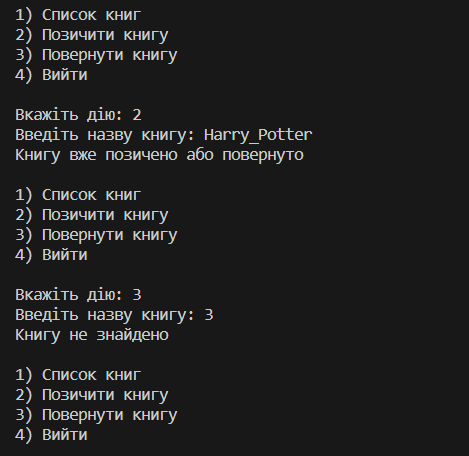


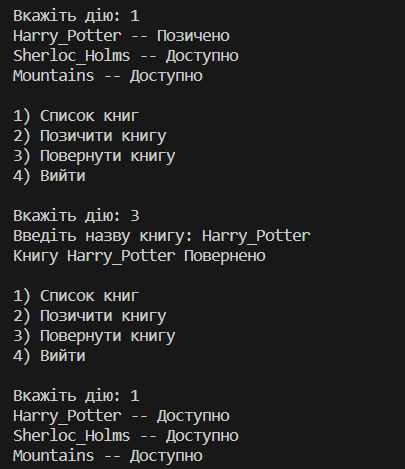
* Task 5: Lab# programming: VNS Lab 7

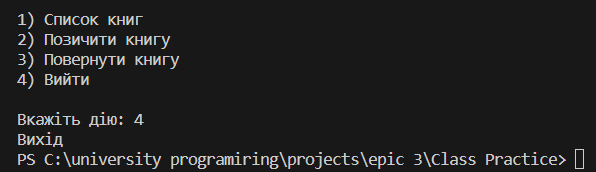


* Task 6: Practice# programming: Class Practice Task

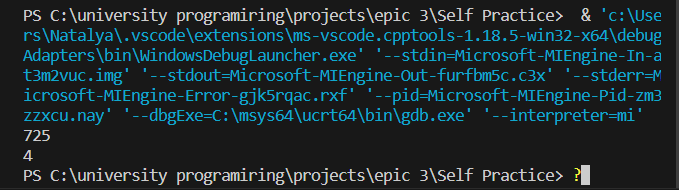








* Task 7: Practice# programming: Self Practice Task



**Посилання на pull request**

[**https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/505/files**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/505/files)

**Висновки:** Навчився створювати функції, застосовувати рекурсію та перевантаження функцій, використовувати вкладені цикли, зрозумів, що таке дебагер