Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 2, 3, 7**

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

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

***Виконав:***

студент групи ШІ-11

Рєвков Володимир

# **Тема роботи:**

Ознайомлення з темою «Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія», виконання лабораторних робіт №2, 3, 7 та практичних завдань.

# **Мета роботи:**

Ознайомитись з темою «Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія», виконати лабораторні роботи №2, 3, 7 та практичні завдання.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли
* Тема №2: Вкладені цикли
* Тема №3: Функції
* Тема №4: Перевантаження функції
* Тема №5: Рекурсія

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли
  + Джерела Інформації
    - Стаття.

<https://www.geeksforgeeks.org/cpp-loops/>

* + Що опрацьовано:
    - Контент зв’язаний з циклами
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 19.11.2023
  + Звершення опрацювання теми: 19.11.2023
* Тема №2: Вкладені цикли
  + Джерела Інформації:
    - Відео.

<https://youtu.be/70CkVC2vVCQ?feature=shared>

* + - Стаття.

<https://www.geeksforgeeks.org/nested-loops-in-c-with-examples-2/>

* + Що опрацьовано:
    - Контент зв’язаний з вкладеними циклами
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 19.11.2023
  + Звершення опрацювання теми: 19.11.2023
* Тема №3: Функції
  + Джерела Інформації:
    - Відео.

<https://youtu.be/BGmDRQzY4CA?feature=shared>

* + - Стаття.

<https://www.geeksforgeeks.org/functions-in-cpp/>

* + Що опрацьовано:
    - Контент зв’язаний з функціями
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 19.11.2023
  + Звершення опрацювання теми: 19.11.2023
* Тема №4: Перевантаження функції
  + Джерела Інформації:
    - Відео.

<https://youtu.be/5OUttXvf6hw?feature=shared>

* + - Стаття.

<https://www.geeksforgeeks.org/function-overloading-c/>

* + Що опрацьовано:
    - Контент зв’язаний з перезавантаженням функції
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 19.11.2023
  + Звершення опрацювання теми: 19.11.2023
* Тема №5: Рекурсія
  + Джерела Інформації:
    - Відео.

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

* + - Стаття.

<https://www.geeksforgeeks.org/cpp-recursion/>

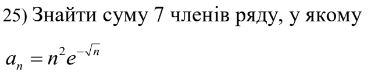
* + Що опрацьовано:
    - Контент зв’язаний з рекурсією
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 19.11.2023
  + Звершення опрацювання теми: 19.11.2023

# **Виконання роботи:**

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

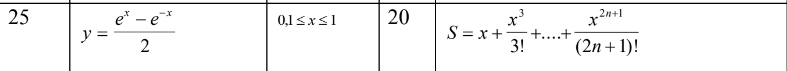
Завдання №1: Лабораторна робота №2

* Варіант завдання №25
* Деталі завдання: *Обчислення суми членів ряду.*



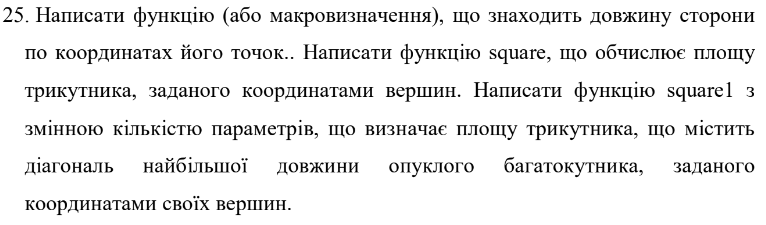
Завдання №2: Лабораторна робота №3

* Варіант завдання №25
* Деталі завдання: *Обчислення функції використовуючи її розклад в степеневий ряд.*



Завдання №3: Лабораторна робота №7

* Варіант завдання № 25
* Деталі завдання: *Обчислення довжини сторони за координатами і тд.*



Завдання №4: Практичне завдання №1

* Варіант завдання «Library Management App»
* Деталі завдання: *Створення застосунку для керування бібліотекою.*

**

Завдання №5: Практичне завдання №2

* Варіант завдання «Patterns Output App»
* Деталі завдання: *Вивід фігур зірочками.*

Програма просить користувача вибрати фігуру(її номер), після запитує, яку фігуру вивести(пусту чи заповнену) та виводить її. Програма працює доти, доки користувач не вибере опцію «Exit».

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

* Програма №2: *Обчислення функції використовуючи її розклад в степеневий ряд.*
* Блок-схема:

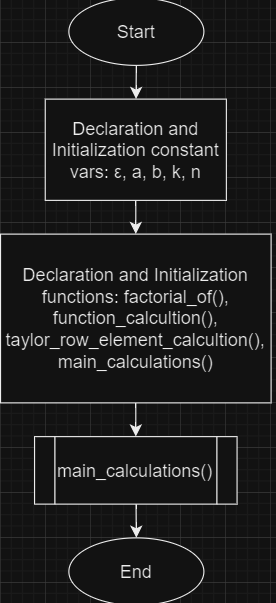
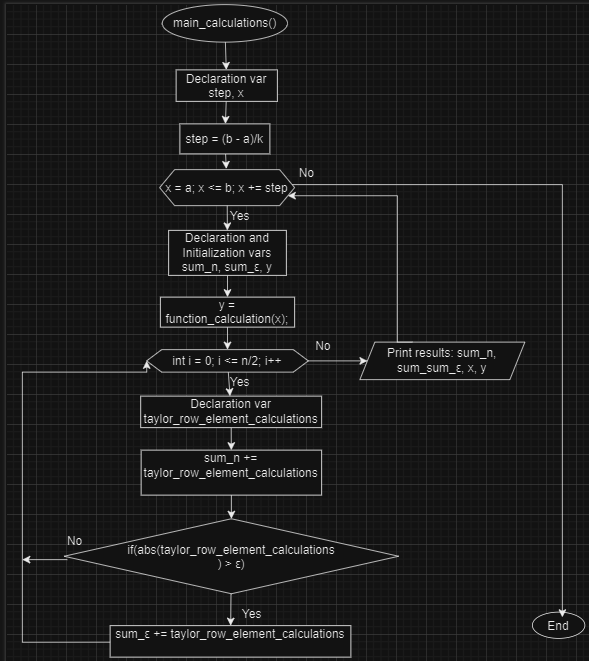
 

Figure 1: Main function Figure 2: main\_calculations function

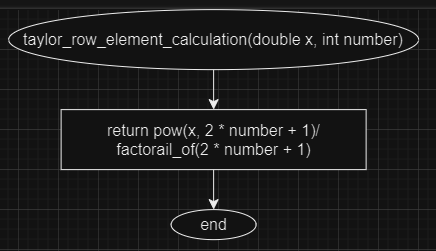
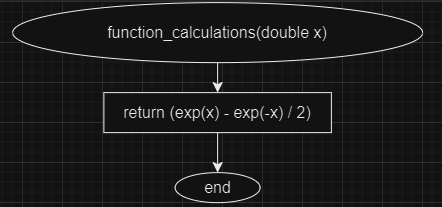
 

Figure 3: function\_calculations function Figure 4: taylor\_row\_calculations function

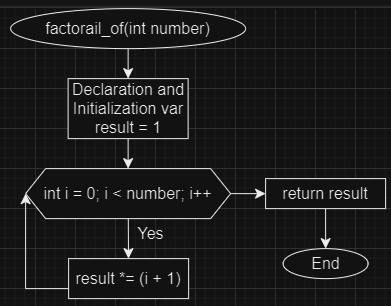


Figure 5: factorial\_of function

* Планований час на реалізацію: 1 година
* Важливі деталі для врахування в імплементації:

У зв’язку з тим, що ця програма здалась мені найбільш складною,

я розробив для неї блок-схему.

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

Завдання №1: Лабораторна робота №2

* Варіант завдання №25
* Деталі завдання: *Обчислення суми членів ряду.*

#include<iostream>

#include<cmath>

using namespace std;

double GetSum(int n)

{

double sum = 0;

    if(n == 0)

    {

        return sum;

    }

    sum = pow(n, 2)\*exp(-sqrt(n));

    n--;

    return sum + GetSum(n);

}

int main()

{

    const int n = 7;

    cout << "Sum of " << n << " elements is: " << GetSum(n);

}

<https://github.com/volodymyr-rievkov/VolodymyrR_Reprository/blob/main/Epic%203/Epic3Lab2T1/vns_lab_2_task_1_variant_25_volodymyr_rievkov.cpp>

Завдання №2: Лабораторна робота №3

* Варіант завдання №25
* Деталі завдання: *Обчислення функції використовуючи її розклад в степеневий ряд.*

using namespace std;

#include<iostream>

#include<cmath>

const double ε = 0.0001;

const double a = 0.1;

const int b = 1;

const int k = 10;

const int n = 20;

int factorial\_of(int number)

{

    int result = 1;

    for(int i = 0; i < number; i++)

    {

        result \*= (i + 1);

    }

    return result;

}

double function\_calculation(double x)

{

    return (exp(x)-exp(-x))/2;

}

double taylor\_row\_element\_calculation(double x, int number)

{

    return pow(x, 2 \* number + 1)/factorial\_of(2 \* number + 1);

}

void main\_calculations()

{

    double step = (b - a)/k;

    for(double x = a; x <= b; x += step)

    {

        double sum\_n = 0;

        double sum\_ε = 0;

        double y = function\_calculation(x);

        for(int i = 0; i <= n/2; i++)

        {

            double taylor\_row\_element\_calculations = taylor\_row\_element\_calculation(x, i);

            sum\_n += taylor\_row\_element\_calculations;

            if(abs(taylor\_row\_element\_calculations) > ε)

            {

                sum\_ε += taylor\_row\_element\_calculations;

            }

        }

        cout << "X = " << x << ",   SN = " << sum\_n << ",   SE = " << sum\_ε << ",   Y = " << y << endl;

    }

}

int main()

{

    main\_calculations();

}

<https://github.com/volodymyr-rievkov/VolodymyrR_Reprository/blob/main/Epic%203/Epic3Lab3T1/vns_lab_3_task_1_variant_25_volodymyr_rievkov.cpp>

Завдання №3: Лабораторна робота №7

* Варіант завдання №25
* Деталі завдання: *Обчислення довжини сторони за координатами і тд.*

#include<iostream>

#include<vector>

#include<cmath>

using namespace std;

struct  point

{

    int x;

    int y;

};

double get\_length\_by\_points(point A, point B)

{

    return sqrt(pow(B.x- A.x, 2) + pow(B.y - A.y, 2));

}

void print\_length\_by\_point(point A, point B)

{

    cout << "Length between points (" << A.x << ", " << A.y << "), (" << B.x << ", " << B.y << ") = " << get\_length\_by\_points(A, B) << endl;

}

double get\_triangle\_square\_by\_points(point A, point B, point C)

{

    double a = get\_length\_by\_points(A, B);

    double b = get\_length\_by\_points(A, C);

    double c = get\_length\_by\_points(B, C);

    double half\_perimeter = (a + b + c)/2;

    return sqrt(half\_perimeter \* (half\_perimeter - a) \* (half\_perimeter - b) \* (half\_perimeter - c));

}

void print\_triangle\_square\_by\_points(point A, point B, point C)

{

    cout << "Square of triangle on points (" << A.x << ", " << A.y << "), (" << B.x << ", " << B.y << "), (" << C.x << ", " << C.y << ") = " << get\_triangle\_square\_by\_points(A, B, C) << endl;

}

double get\_triangle\_square\_by\_max\_diagonal(vector<point>points)

{

    int size = points.size();

    point A, B, C;

    double max\_diagonal = 0;

    for(int i = 0; i < size; i++)

    {

        for(int y = 0; y < size; y++)

        {

            double length = get\_length\_by\_points(points[i], points[y]);

            if(length > max\_diagonal)

            {

                max\_diagonal = length;

                A = points[i];

                B = points[y - 1];

                C = points[y];

            }

        }

    }

    return get\_triangle\_square\_by\_points(A, B, C);

}

void print\_triangle\_square\_by\_max\_diagonal(vector<point>points)

{

    cout << "Square of triangle by max diagonal on points ";

    for(int i = 0; i < points.size(); i++)

    {

        cout << "(" << points[i].x << ", " << points[i].y << ")";

        if(i != points.size() - 1)

        {

            cout << ", ";

        }

    }

    cout << " = " << get\_triangle\_square\_by\_max\_diagonal(points) << endl;

}

int main()

<https://github.com/volodymyr-rievkov/VolodymyrR_Reprository/blob/main/Epic%203/Epic3Lab7T1/vns_lab_7_task_1_variant_25_volodymyr_rievkov.cpp>

Завдання №4: Практична робота №1

* Варіант завдання «Library Management App»
* Деталі завдання: *Створення застосунку для керування бібліотекою.*

#include<iostream>

#include<string>

#include<vector>

std::vector<std::string> libraryInventory = {"Robinson Crusoe D. Dafoe",

                                                "Gulliver's Travels J. Swift",

                                                "Frankenstein M. Shelley",

                                                "Little Women L. May",

                                                "Three man in a Boat J. Jerome"};

std::vector<std::string> libraryInventoryavailibility = {"Availible", "Availible", "Availible", "Availible", "Availible"};

char action;

bool libraryExit = false;

int bookNumber;

bool isContinued;

void PrintLibraryInventoryandAvailibility(std::vector<std::string> books, std::vector<std::string> availibility)

{

    std::cout << "----------Library Inventory----------" << std::endl;

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

    {

        for(int y = 0; y < size(availibility); y++)

        {

            if(i == y)

            {

                std::cout << i + 1 <<") " << books[i] << " - " << availibility[y] << std::endl;

            }

        }

    }

}

void GetAction()

{

    PrintLibraryInventoryandAvailibility(libraryInventory, libraryInventoryavailibility);

    std::cout << "Take book(T) \nReturn book(R) \nExit library(E)\nEnter action: ";

    std::cin >> action;

}

void GetNumberOfBook()

{

    PrintLibraryInventoryandAvailibility(libraryInventory, libraryInventoryavailibility);

    std::cout << "Enter number of book: ";

    std::cin >> bookNumber;

}

void TakeBook()

{

    for(int i = 0; i < size(libraryInventoryavailibility); i++)

    {

        if(bookNumber == i + 1 && libraryInventoryavailibility[i] == "Availible")

        {

            libraryInventoryavailibility[i] = "Non Availible";

        }

    }

}

void ReturnBook()

{

    for(int i = 0; i < size(libraryInventoryavailibility); i++)

    {

        if(bookNumber == i + 1 && libraryInventoryavailibility[i] == "Non Availible")

        {

            libraryInventoryavailibility[i] = "Availible";

        }

    }

}

void ExitLibrary()

{

    libraryExit = true;

    std::cout << "GoodBye :)" << std::endl;

}

void IsContinued()

{

    std::cout << "Do you want to contine(Y/N): ";

    std::cin >> action;

    if(action == 'Y')

    {

        isContinued = true;

    }

    else

    {

        isContinued = false;

    }

}

int main()

{

    while(!libraryExit)

    {

    mainMenu:

        GetAction();

        switch(action)

        {

            case 'T':

                do

                {

                    GetNumberOfBook();

                    if((bookNumber < 1 || bookNumber > size(libraryInventory)) || libraryInventoryavailibility[bookNumber - 1] == "Non Availible")

                    {

                        goto mainMenu;

                    }

                    TakeBook();

                    IsContinued();

                }while(isContinued);

                goto mainMenu;

                break;

            case 'R':

                do

                {

                    GetNumberOfBook();

                    if((bookNumber < 1 || bookNumber > size(libraryInventory)) || libraryInventoryavailibility[bookNumber - 1] == "Availible")

                    {

                        goto mainMenu;

                    }

                    ReturnBook();

                    IsContinued();

                }while(isContinued);

                goto mainMenu;

                break;

            case 'E':

                ExitLibrary();

                break;

            default:

                goto mainMenu;

                break;

        }

    }

}

<https://github.com/volodymyr-rievkov/VolodymyrR_Reprository/blob/main/Epic%203/Practise1LibraryApp/practise_task_1_volodymyr_rievkov.cpp>

Завдання №5: Практична робота №2

* Варіант завдання «Patterns Output App»
* Деталі завдання: *Вивід фігур зірочками.*

#include<iostream>

#include<vector>

#include<functional>

using namespace std;

int sideSize;

bool lever = true;

int action;

void GetSideSize()

{

    cout << "Enter size: ";

    cin >> sideSize;

}

void ChoosePattern()

{

    do

    {

        cout << " 1 - Square\n 2 - Rectangle\n 3 - Triangle\n 4 - Pyramid\n 5 - Exit\nEnter number: ";

        cin >> action;

    } while (action != 1 && action != 2 && action != 3 && action != 4 && action != 5 && action != 6);

}

bool IsFilled()

{

    char choice;

    bool isFilled;

    do

    {

        cout << "Do you want to fill pattern(y/n): ";

        cin >> choice;

    } while (choice != 'y' && choice != 'n');

    if(choice == 'y')

    {

        isFilled = true;

    }

    else

    {

        isFilled = false;

    }

    cout << endl;

    return isFilled;

}

void PrintSquare()

{

    GetSideSize();

    if(IsFilled())

    {

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

        {

            for(int y = 0; y < sideSize; y++)

            {

                cout << "\* ";

            }

            cout << endl;

        }

    }

    else

    {

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

        {

            for(int y = 0; y < sideSize; y++)

            {

                if(y == 0 || i == 0 || y == sideSize - 1 || i == sideSize - 1)

                {

                    cout << "\* ";

                }

                else

                {

                    cout << "  ";

                }

            }

            cout << endl;

        }

    }

    cout << endl;

}

void PrintRectangle()

{

    GetSideSize();

    int doubleSidesize = sideSize \* 2;

    if(IsFilled())

    {

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

        {

            for(int y = 0; y < doubleSidesize; y++)

            {

                cout << "\* ";

            }

            cout << endl;

        }

    }

    else

    {

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

        {

            for(int y = 0; y < doubleSidesize; y++)

            {

                if(y == 0 || i == 0 || y == doubleSidesize - 1 || i == sideSize - 1)

                {

                    cout << "\* ";

                }

                else

                {

                    cout << "  ";

                }

            }

            cout << endl;

        }

    }

    cout << endl;

}

void PrintTriangle()

{

    GetSideSize();

    if(IsFilled())

    {

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

        {

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

            {

                cout << "\* ";

            }

            cout << endl;

        }

    }

    else

    {

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

        {

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

            {

                if (y == i || i == sideSize - 1 || y == 0)

                {

                    cout << "\* ";

                }

                else

                {

                    cout << "  ";

                }

            }

            cout << endl;

        }

    }

    cout << endl;

}

void PrintPyramid()

{

    GetSideSize();

    int doubleSidesize = sideSize \* 2;

    if(IsFilled())

    {

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

        {

            int spaces = (sideSize - (i + 1));

            for(int y = 0; y < doubleSidesize - 1; y++)

            {

                if(spaces > 0 || y >= sideSize + i)

                {

                    cout << "  ";

                    spaces--;

                }

                else

                {

                    cout << "\* ";

                }

            }

            cout << endl;

        }

    }

    else

    {

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

        {

            for(int y = 0; y < doubleSidesize - 1; y++)

            {

                if(y == sideSize - i - 1 || y == sideSize + i - 1 || i == sideSize - 1)

                {

                    cout << "\* ";

                }

                else

                {

                    cout << "  ";

                }

            }

            cout << endl;

        }

    }

    cout << endl;

}

void Exit()

{

    lever = false;

}

vector<function<void()>>patterns {PrintSquare, PrintRectangle, PrintTriangle, PrintPyramid, Exit};

int main()

<https://github.com/volodymyr-rievkov/VolodymyrR_Reprository/blob/main/Epic%203/Practise2PatternsOutput/practise_task_2_volodymyr_rievkov.cpp>

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

Завдання №1: Лабораторна робота №2

* Варіант завдання №25
* Деталі завдання: *Обчислення суми членів ряду.*



Figure 6: Вивід результату ЛБ №2

Час затрачений на виконання завдання: 30 хвилин.

Завдання №2: Лабораторна робота №3

* Варіант завдання №25
* Деталі завдання: *Обчислення функції використовуючи її розклад в степеневий ряд.*

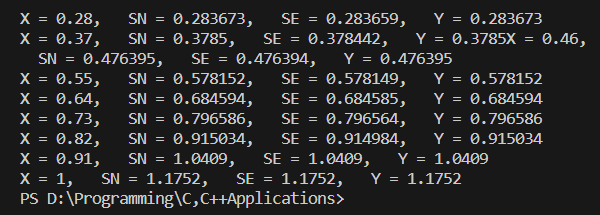


Figure 7: Вивід результату ЛБ №3

Час затрачений на виконання завдання: 6 годин.

Завдання №3: Лабораторна робота №7

* Варіант завдання №25
* Деталі завдання: *Обчислення довжини сторони за координатами і тд.*

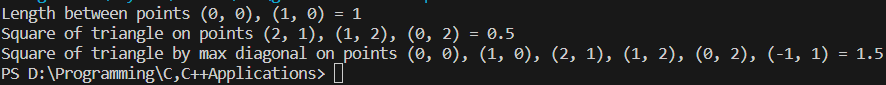


Figure 8: Вивід результату ЛБ №7

Час затрачений на виконання завдання: 3 години.

Завдання №4: Практична робота №1

* Варіант завдання «Library Management App»
* Деталі завдання: *Створення застосунку для керування бібліотекою.*

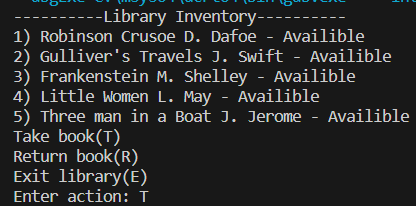
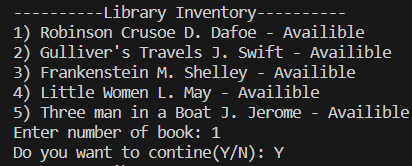
** 

Figure 9: Вибрано опцію T-Take Figure 10: Обрано книжку №1 та продовжити

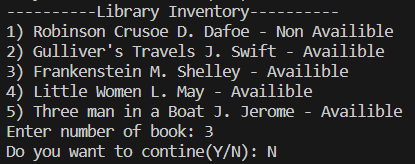
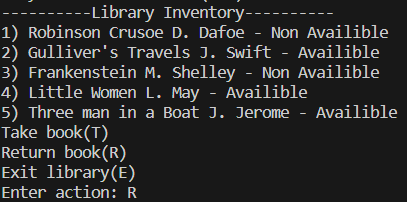
 

Figure 11: Доступність книжки змінилась Figure 12: Вибрано опцію R-Return

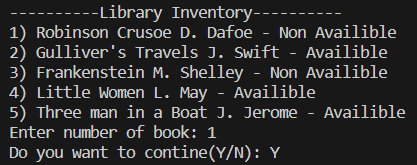
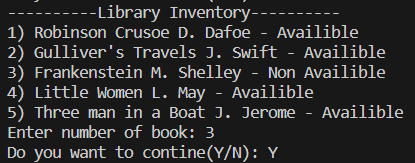
 

Figure 13: Обрано книжку №1 та продовжити Figure 14: Доступність книжки змінилась

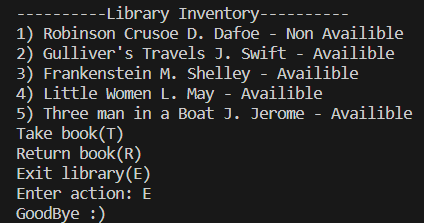


Figure 15: Вибрано опцію E-Exit

Час затрачений на виконання завдання: 1 година.

Завдання №5: Практична робота №2

* Варіант завдання «Patterns Output App»
* Деталі завдання: *Вивід фігур зірочками.*

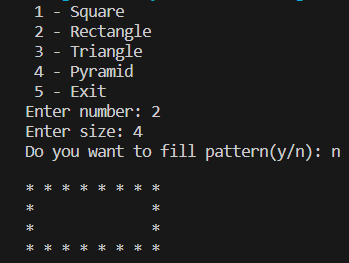
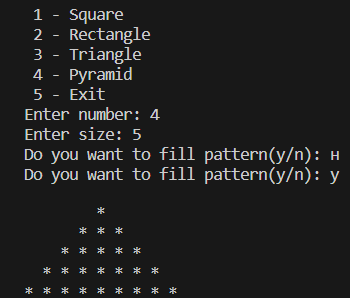
 

Figure 16: Вивід пустого прямокутника 4\*8 Figure17: Вивід заповненої піраміди 5\*9

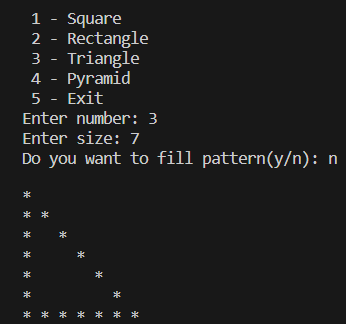
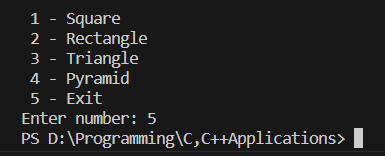
 

Figure 18: Вивід пустого трикутника 7\*7 Figure 19: Вибрано опцію 5-Exit

Час затрачений на виконання завдання: 3 години.

# **Висновки:**

На роботу з завданнями Epic 3 пішло багато часу та терпіння, але не дарма, адже я вивчив багато нового. Наприклад: 1)Завдяки чудесним завданням з ВНС покращив свої знання в алгебрі та геометрії, 2)Пригадав рекурсивну функцію, 3)Навчився створювати функції з змінною кількістю параметрів, 4)Пригадав структуру, 5)Також пригадав функціонал вкладених циклів, завдяки завданню №5 «Patterns Output App».