Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**Звіт**

**про виконання лабораторних та практичних робіт блоку № 3**

На тему: «Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції з змінною кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції.»

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

до:

ВНС Лабораторної Роботи № 2

ВНС Лабораторної Роботи № 3

ВНС Лабораторної Роботи № 7

Практичних Робіт № 3

***Виконав:***

студент групи ШІ-14

Кроляк Роман Романович

# **Тема роботи:**

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

# **Мета роботи:**

Освоїти роботу з циклами, функціями, перевантаженням функцій та написанням рекурсивних функцій.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли
* Тема №2: Функції
* Тема №3: Перевантажування функцій
* Тема №4: Рекурсивні функції

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли
  + Джерела Інформації
    - <https://www.w3schools.com/cpp/cpp_while_loop.asp>
    - <https://www.w3schools.com/cpp/cpp_do_while_loop.asp>
    - <https://www.w3schools.com/cpp/cpp_for_loop.asp>
    - <https://www.w3schools.com/cpp/cpp_break.asp>
  + Що опрацьовано:
    - Вивчення роботи із циклами for, while, do while, а також використання операторів break і continue.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.11.2022
  + Звершення опрацювання теми: 19.11.2022
* Тема №2: Функції
  + Джерела Інформації:
    - <https://www.w3schools.com/cpp/cpp_functions.asp>
    - <https://www.w3schools.com/cpp/cpp_function_param.asp>
    - <https://www.w3schools.com/cpp/cpp_function_default.asp>
    - <https://www.w3schools.com/cpp/cpp_function_multiple.asp>
    - <https://www.w3schools.com/cpp/cpp_function_return.asp>
    - <https://www.w3schools.com/cpp/cpp_function_reference.asp>
    - <https://www.w3schools.com/cpp/cpp_function_array.asp>
  + Що опрацьовано:
    - Що таке функції
    - Для чого їх використовують
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.11.2022
  + Звершення опрацювання теми: 23.11.2022
* Тема №3: Перевантажування функцій
  + Джерела Інформації:
    - <https://www.w3schools.com/cpp/cpp_function_overloading.asp>
  + Що опрацьовано:
    - Що таке перевантаження функцій
    - Для чого його застосовують
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2022
  + Звершення опрацювання теми: 27.11.2022
* Тема №4: Рекурсивні функції
  + Джерела Інформації:
    - <https://www.w3schools.com/cpp/cpp_functions_recursion.asp>
  + Що опрацьовано:
    - Що таке рекурсивна функція
  + Статус: Ознайомлений
  + Початок опрацювання теми: 28.11.2022
  + Звершення опрацювання теми: 30.11.2022

# **Виконання роботи:**

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

Завдання №1 VNS Lab 2

* Варіант завдання: 2
* Знайти суму цілих додатніх парних чисел, менших 100.

Завдання №2 VNS Lab 3

* Варіант завдання: 2
* Для х, що змінюється від a до b з кроком (b-a)/10, обчислити функцію f(x), використовуючи її розклад в степеневий ряд для заданого n та для заданої точності ε=0.0001. Обчислення будуть найбільш ефективними, якщо кожен член суми обчислювати за загальною формулою an=(x, n).
* 
* Алгоритм розв’язання завдання зводиться до трьох циклів, причому два з них вкладені в третій. Внутрішні цикли підсумують доданки при фіксованому параметрі x, один для заданого n, інший для заданої точності ε. Зовнішній цикл організує зміну параметра х.

Завдання №3 VNS Lab 7 Task 1

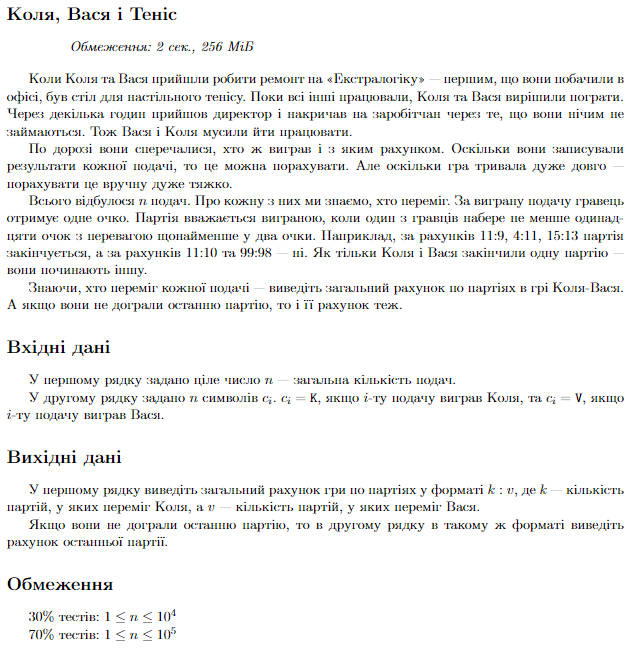
* Варіант завдання: 2
* Написати функцію mult зі змінною кількістю параметрів, що знаходить добуток чисел типу float. Написати викликаючу функцію main, що звертається до функції mult не менше трьох разів з кількістю параметрів 3, 7, 11.

Завдання №4 Class Practice Task

* Перерахувати всі книги.
* 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.
* Вимоги:

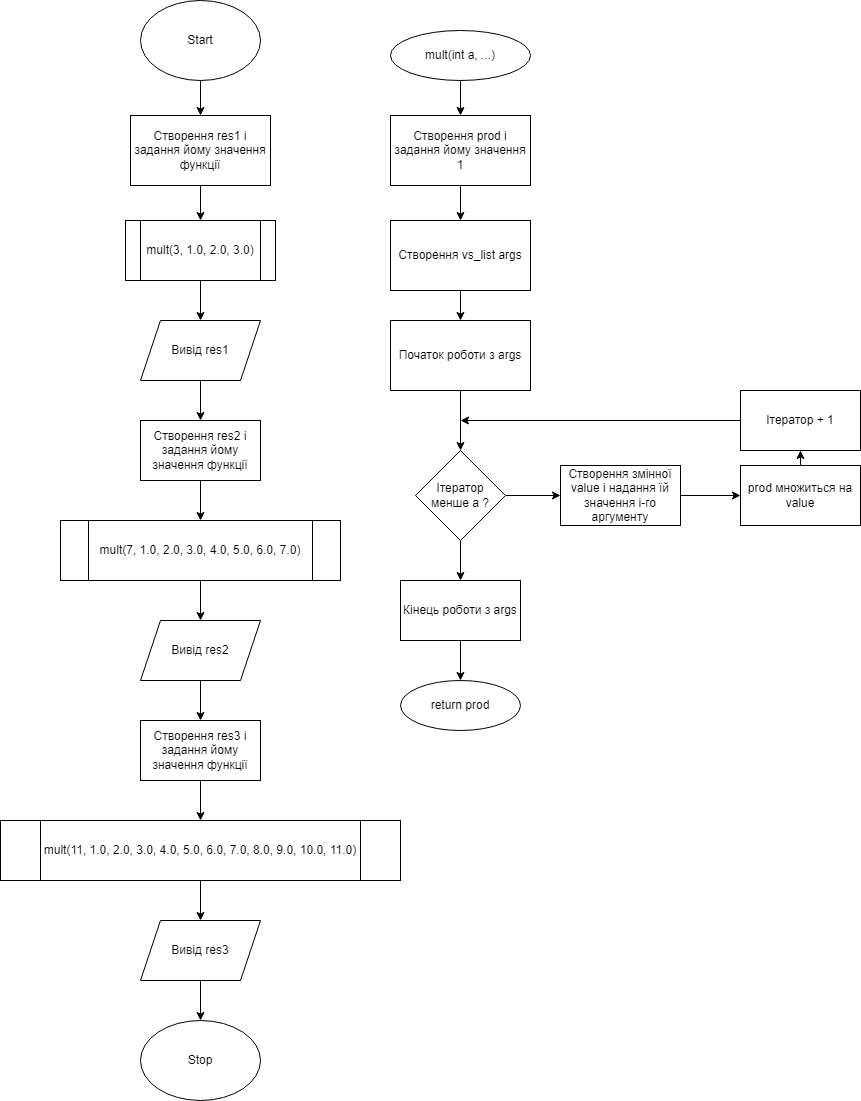
1. while: продовжувати працювати, доки користувач не вирішить вийти.
2. do while: Після кожної операції (позичити, повернути, перерахувати) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.
3. for: список усіх книг за допомогою циклу.
4. for each: перевірити наявність кожної книги.
5. goto: якщо користувач вводить неправильний вибір, використовуйте goto, щоб перенаправити його до головного меню.

Завдання №5 Self Practice Task

* [https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/13](https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/13#mySolutions)
* 

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

Програма №3 VNS Lab 7 Task 1



*Блок схема до програми №3*

* Планований час на реалізацію: 30 хв

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

## Конфігурація середовища незмінна

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

Завдання №1 VNS Lab 2

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/672/files#diff-8f49e3845717d03a3a51f78f558f1e7cfd2c24b8649c4b983ef6e624e76b866a

#include <iostream>

using namespace std;

int main() {

int sum = 0;

for (int i = 0; i < 100; i+=2) sum+=i;

cout << "Sum even (1-100): " << sum;

return 0;

}

Завдання №2 VNS Lab 3

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/672/files#diff-3ff7d49b97a45fcf87c1f4a0664f1b1aa49b3e7ea4326d3f66805fbaec4d7f1b

#include <iostream>

#include <cmath>

using namespace std;

int main() {

double step = (9.0\*M\_PI/5.0 - M\_PI/5.0)/10.0;

for (double x = M\_PI/5; x <= 9\*M\_PI/5; x += step) {

double y = -log(fabs(2\*sin(x/2)));

cout << "X=" << x << " ";

double sum\_a = 0.0;

for (int i = 1; i <= 40; ++i) {

sum\_a += cos(i\*x)/i;

}

cout << "SN=" << sum\_a << " ";

double sum\_b = 0.0;

double a\_b = cos(x);

for (int i = 2; fabs(a\_b) >= 0.0001; ++i) {

sum\_b += a\_b;

a\_b = cos(i\*x)/i;

}

cout << "SE=" << sum\_b << " ";

cout << "Y=" << y << endl;

}

return 0;

}

Завдання №3 VNS Lab 7 Task 1

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/672/files#diff-04e6fd2bdd0816054598e27e47c4d971e4df55b6f8b432f1bb5e7cd61aeb81c7

#include <iostream>

#include <cstdarg>

using namespace std;

float mult(int a, ...){

float prod = 1;

va\_list args;

va\_start(args, a);

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

{

float value = va\_arg(args, double);

prod \*= value;

}

va\_end(args);

return prod;

}

int main(){

float res1 = mult(3, 1.0, 2.0, 3.0);

cout << "Prod of 3: " << res1 << endl;

float res2 = mult(7, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0);

cout << "Prod of 7: " << res2 << endl;

float res3 = mult(11, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0);

cout << "Prod of 11: " << res3 << endl;

return 0;

}

Завдання №4 Class Practice Task

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/672/files#diff-f0671c307029672246e33a8b4b5e0a33e42e40a93dfafc015e77b077dfbc59de

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main(){

bool cont = true;

bool other\_op;

vector<string> book\_names = **{**"Harry Potter", "Kobzar", "The Churchill Factor", "Shadows of Forgotten Ancestors", "1984", "Treasure Island", "Miss Peregrine's Home for Peculiar Children"**}**;

bool book\_availability[book\_names.size()];

for (bool &avaib : book\_availability) avaib = true;

while(cont){

do

{

int choice;

cout << "What do you want to do? \n1. Print names of books \n2. Take book \n3. Return book\n";

choose\_op:

cout << "Your choice:";

cin >> choice;

switch (choice)

{

case 1:

{

for (int i = 0; i < book\_names.size(); i++)

cout << i+1 << ". " << book\_names[i] << endl;

break;

}

case 2:

{

vector<int> available\_i;

cout << "Available books:\n";

int i = 0;

int j = 0;

for (bool avaib : book\_availability){

if (avaib) {

cout << j+1 << ". " << book\_names[i] << endl;

available\_i.push\_back(i);

j++;

}

i++;

}

int book\_choice\_take;

cout << "What book do you want to take?\n";

choose\_take:

cout << "Your choice:";

cin >> book\_choice\_take;

if (book\_choice\_take <= 0 || book\_choice\_take > j+1)

{

cout << "Invalid choice. Choose 1-" << j+1;

goto choose\_take;

}

book\_availability[available\_i[book\_choice\_take-1]] = false;

break;

}

case 3:

{

vector<int> taken\_i;

cout << "Taken books:\n";

int i = 0;

int j = 0;

for (bool avaib : book\_availability){

if (!avaib) {

cout << j+1 << ". " << book\_names[i] << endl;

taken\_i.push\_back(i);

j++;

}

i++;

}

int book\_choice\_return;

cout << "What book do you want to return?\n";

choose\_return:

cout << "Your choice:";

cin >> book\_choice\_return;

if (book\_choice\_return <= 0 || book\_choice\_return > j+1)

{

cout << "Invalid choice. Choose 1-" << j+1;

goto choose\_return;

}

book\_availability[taken\_i[book\_choice\_return-1]] = true;

break;

}

default:

{

cout << "Invalid operation. Choose 1-3";

goto choose\_op;

}

}

char an\_op;

cout << "Do another operation? (y/n):";

choose\_an\_op:

cin >> an\_op;

switch (an\_op)

{

case 'y':

other\_op = true;

break;

case 'n':

other\_op = false;

break;

default:

cout << "Invalid input. Choose y or n";

goto choose\_an\_op;

}

} while (other\_op);

char quit;

cout << "Quit program? (y/n):";

choose\_quit:

cin >> quit;

switch (quit)

{

case 'y':

cont = false;

break;

case 'n':

cont = true;

break;

default:

cout << "Invalid input. Choose y or n";

goto choose\_quit;

}

}

return 0;

}

Програма №5 Self Practice Task

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/672/files#diff-cf3cca8d736bcde21c598ba358bda5886494d0071f75be06562b6065dbc05f55

#include <iostream>

#include <string>

using namespace std;

int main() {

int n;

string s;

cin >> n;

cin >> s;

int points\_K, points\_V, games\_K, games\_V;

points\_K = points\_V = games\_K = games\_V = 0;

bool IncorrectInput = false;

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

(s[i] == 'K') ? points\_K++ : (s[i] == 'V') ? points\_V++ : IncorrectInput = true;

if (IncorrectInput) return -1;

if (points\_K >= 11 && points\_K >= points\_V+2) {

games\_K++;

points\_K = points\_V = 0;

}

else if (points\_V >= 11 && points\_V >= points\_K+2) {

games\_V++;

points\_K = points\_V = 0;

}

}

cout << games\_K << ":" << games\_V << endl;

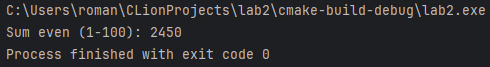
if (points\_K != 0 || points\_V != 0) cout << points\_K << ":" << points\_V;

return 0;

}

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

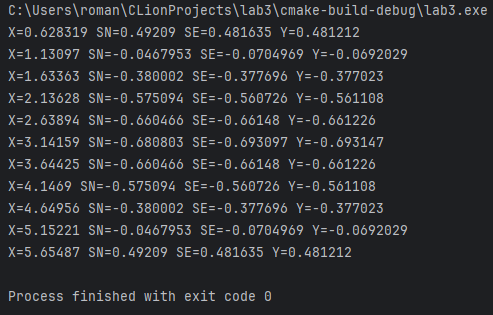
Завдання №1 VNS Lab 2



*Результат виконання програми №1*

Час затрачений на виконання завдання: 5 хв

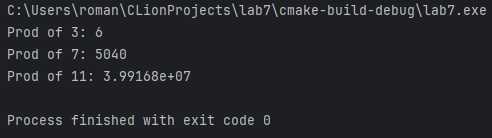
Завдання №2 VNS Lab 3



*Результат виконання програми №2*

Час затрачений на виконання завдання: 60 хв

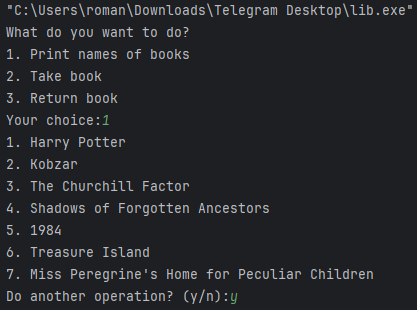
Завдання №3 VNS Lab 7 Task 1

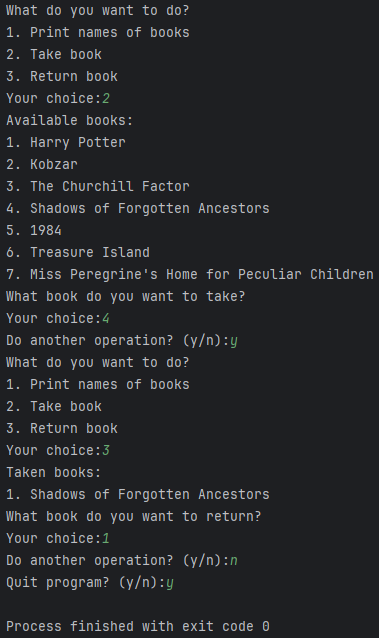


*Результат виконання програми №3*

Час затрачений на виконання завдання: 30 хв

Завдання №4 Class Practice Task

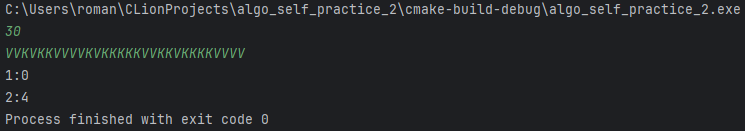




*Результат виконання програми №4*

Час затрачений на виконання завдання: 100 хв

Завдання №5 Self Practice Task



*Результат виконання програми №5*



*Підтвердження коректної роботи програми №5*

Час затрачений на виконання завдання: 30 хв

# **Висновки:**

У третьому етапі я здобув навички роботи з циклами, створення рекурсивних функцій та перегрузки функцій.