Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 2**

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

***з розділу***: «Лінійні та розгалужені алгоритми. Умовні оператори. Константи, змінні»

***Виконав:***

студент групи ШІ-14

Кисіль Сергій Дмитрович

# **Тема роботи:**

# Цикли for(+foreach), while, do while, вкладенні цикли, функції та типи повернення функцій, функції типу void, перевантаження функцій, рекурсивні функції

# **Мета роботи:**

Ознайомитись з різними типами циклів, такими як: while, do while, for(+foreach для масивів), опробувати застосування вкладених циклів. Ознайомитись з функціями, параметрами та аргументами функцій, типами повернення функцій, зокрема void, навчитись перевантажувати функції для різних типів та різної кількості параметрів.

Завдання:

⦁ Epic 3 Task 1 - Theory Education Activities

⦁ Epic 3 Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)

⦁ Epic 3 Task 3 - Lab# programming: VNS Lab 3

⦁ Epic 3 Task 4 - Lab# programming: VNS Lab 7

⦁ Epic 3 Task 5 - Practice# programming: Class Practice Task

Epic 3 Task 6 - Practice# programming: Self Practice Task

⦁ Epic 3 Task 7 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

⦁ Epic 3 Task 8 - Results Evaluation and Release

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

⦁ Тема №1: Цикли while, do while, for i foreach

⦁ Тема №2: Функції та різновиди типів повернень функції

⦁ Тема №3: Еліпсис(функції з нефіксованою кількістю параметрів)

⦁ Тема №4: Рекурсія

⦁ Тема №5: Перевантаження функцій

⦁ Тема №6: Типи передачі аргументів у функції

1. Індивідуальний план опрацювання теорії:

Тема №1: Цикли while, do while, for i foreach

* + Джерела Інформації:

<https://acode.com.ua/urok-70-tsykl-while/>

<https://acode.com.ua/urok-71-tsykl-do-while/>

<https://acode.com.ua/urok-72-tsykl-for/>

Team meats

Одногрупники та однопотоківці

* + Що опрацьовано:

Цикли while та do while і різниця між ними

Використання for як більш короткого аналога до while/do while

Модифікований цикл for(foreach) для роботи з масивами

* + Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений
  + Початок опрацювання теми: 01.11.2023
  + Завершення опрацювання теми: 29.11.2023

Тема №2: Функції та різновиди типів повернень функції

* + Джерела Інформації:

Team meats

Стаття: [документація до ВНС Лаб 1](https://drive.google.com/drive/u/6/folders/1rywFoh9PJBgf_vJn5EwSyo0Ywo7BvtI3)

* + Що опрацьовано:

Поняття функції та як їх використовувати

Різні типи повернень в функціях, зокрема void

Параметри та аргументи функцій

* + Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений
  + Початок опрацювання теми: 01.11.2023
  + Завершення опрацювання теми: 01.11.2023

Тема №3: Еліпсис(функції з нефіксованою кількістю параметрів)

* + Джерела Інформації:

Team meats

Одногрупники

Google translate

<https://www.programiz.com/cpp-programming/if-else>

* + Що опрацьовано:

Ознайомлення з еліпсисом – об’єктом для визначення функцій з наперед невідомою кількістю параметрів

Використання макросів va\_list, va\_start і va\_end для доступу до еліпсису

Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений

* + Початок опрацювання теми: 01.11.2023
  + Завершення опрацювання теми: 29.11.2023

Тема №4: Рекурсія

* + Джерела Інформації:

Одногрупники

Практичне заняття

<https://www.programiz.com/cpp-programming/if-else> - (if, else, else if)

<https://www.w3schools.com/cpp/cpp_conditions.asp> - (if, else, else if exercises)

<https://www.w3schools.com/cpp/cpp_switch.asp> - (switch)

<https://www.w3schools.com/cpp/cpp_do_while_loop.asp> - (do while)

<https://www.programiz.com/cpp-programming/for-loop> - (for)

* + Що опрацьовано:

Рекурсивні функції та їх використання

Числа Фібоначчі

Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений

* + Початок опрацювання теми: 01.11.2023
  + Звершення опрацювання теми: 29.11.2023

Тема №5: Перевантаження функцій

* + Що опрацьовано:

Перевантаження функцій для різної кількості параметрів

Перевантаження функцій для різних типів параметрів

* + Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений
  + Початок опрацювання теми: 01.11.2023
  + Звершення опрацювання теми: 29.11.2023

Тема №6: Типи передачі аргументів у функції

* + Що опрацьовано:

Передача за значенням

Передача за посиланням

Передача за адресою

* + Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений
  + Початок опрацювання теми: 01.11.2023
  + Звершення опрацювання теми: 29.11.2023

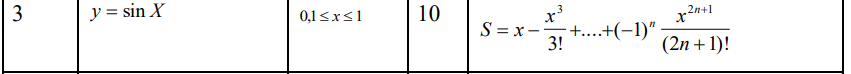
# **Виконання роботи:**

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

* Завдання №1 Лаб# Програмування: VNS Lab 2 (Завдання 1)
* Варіант завдання: 3
* Деталі завдання:Знайти суму цілих додатніх непарних чисел, менших 200.

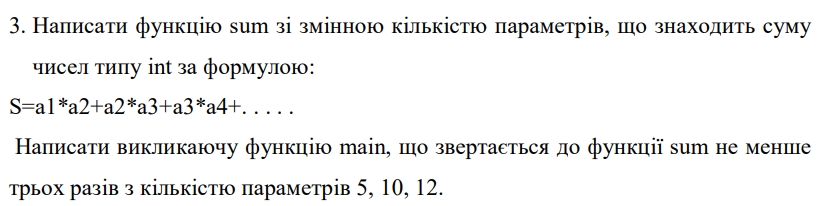
Завдання №2 Лаб# Програмування: VNS Lab 3 (Завдання 1)

Варіант завдання: 3

* Деталі завдання:

Завдання №3 Лаб# Програмування: VNS Lab 7 (Завдання 1)

Варіант завдання: 3

* Деталі завдання:

Завдання №4 Практична# Програмування: Class Practice Task 3

Варіант завдання(відсутній)

* Деталі завдання:

Менеджмент бібліотеки

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.

Вимоги

⁃ while: продовжувати працювати, доки користувач не вирішить вийти.

⁃ do while: Після кожної операції (позичити, повернути, перерахувати) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.

⁃ for: список усіх книг за допомогою циклу for.

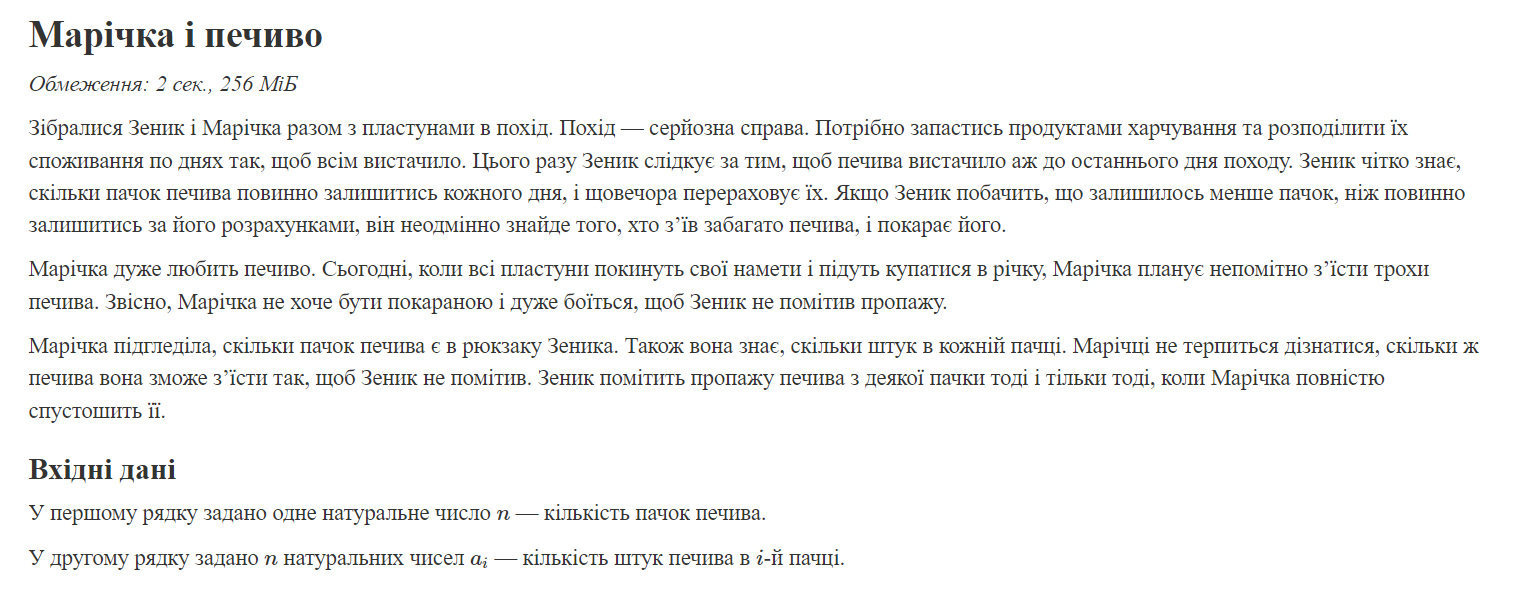
⁃ для кожної: перевірити наявність кожної книги.

⁃ goto: якщо користувач вводить неправильний вибір, використовуйте goto, щоб перенаправити його до головного меню.

Завдання №5 Практична# Програмування: Self Practice Task

Варіант завдання (відсутній)

* Деталі завдання:



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

Програма №1 VNS Lab 2 (Завдання 1)

* Планований час на реалізацію: 10 хвилин.

Програма №2 Лаб# Програмування: VNS Lab 3 (Завдання 1)

* Планований час на реалізацію: 180 хвилин.

Програма №3: Лаб# Програмування: VNS Lab 7 (Завдання 1)

* Блок-схема
* Планований час на реалізацію 60 хвилин.

Програма №4: Практична# Програмування: Class Practice Task

* Блок-схема
* Планований час на реалізацію 1.5 години.

Програма №5: Практична# Програмування: Self Practice Task

* Планований час на реалізацію 20 хвилин.

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

Завдання №1: Лаб# Програмування: VNS Lab 2 (Завдання 1)

#include <iostream>

using namespace std;

int main() {

int sum = 0;

for (int i = 1; i < 200; i += 2) {

sum += i;

}

cout << "Сума цілих додатніх непарних чисел, менших 200: " << sum <<"\n";

return 0;

}

Завдання №2: Лаб# Програмування: VNS Lab 3 (Завдання 1)

#include <iostream>

#include <cmath>

using namespace std;

int fl(int l){

if (l == 0 || l == 1) {

return 1;

} else {

return l \* fl(l - 1);

}

}

int main(){

double a = 0.1, b = 1;

int k = 10;

double c = (b - a) / k;

for (double x = a; x <= b; x += c){

double y = sin(x);

double sumN = x;

double termN = 1;

for (int n = 1; n <= 10; n++){

termN \*= pow(-1, n) \* pow(x, 2\*n + 1)/fl(2\*n + 1);

sumN += termN;

}

double sumE = x;

double termE = 1;

for (int n = 1; fabs(termE) >= 0.0001; n++){

termE \*= pow(-1, n) \* pow(x, 2\*n + 1)/fl(2\*n + 1);

sumE += termE;

}

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

cout << "sn = " << sumN << " ";

cout << "se = " << sumE << " ";

cout << "y = " << y;

cout << endl;

}

return 0;

}

Завдання №3: Лаб# Програмування: VNS Lab 7 (Завдання 1)

#include <iostream>

using namespace std;

int sum(int array[], int length) {

if (length < 2) {

cout << "Потрібно передати принаймні два параметри для обчислення суми." << endl;

return -1;

}

int result = 0;

for (int i = 0; i < length - 1; i++) {

result += array[i] \* array[i + 1];

}

return result;

}

int main() {

int n, a, u;

cin>> n >> u >>a;

int array1[n];

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

array1[i] = i + 1;

}

int result1 = sum(array1, n);

cout << "Результат для " << n << " параметрів: " << result1 << endl;

int array2[u];

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

array2[i] = i + 1;

}

int result2 = sum(array2, u);

cout << "Результат для " << u << " параметрів: " << result2 << endl;

int array3[a];

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

array3[i] = i + 1;

}

int result3 = sum(array3, a);

cout << "Результат для " << a << " параметрів: " << result3 << endl;

return 0;

}

Завдання №4: Практична# Програмування: Class Practice Task

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main() {

vector<string> books = {"Harry Potter", "C++ for kettles", "The Grinch", "Breaking Bad", "a-ba-ba-ga-la-ma-ga"};

vector<bool> available = {true, true, true, true, true};

vector<string> taken;

int n;

do {

cout << "Welcome to Serhii`s library\n";

cout << "1. List all books\n";

cout << "2. Borrow a book\n";

cout << "3. Return a book\n";

cout << "4. List borrowed books\n";

cout << "5. Exit\n";

cout << "Enter your action: ";

cin >> n;

switch (n) {

case 1:

cout << "Available books:\n";

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

if (available[i]) {

cout << books[i] << endl;

}

}

cout << "\n";

break;

case 2:

{

string minus;

cout << "Enter the book you want to borrow: ";

cin.ignore();

getline(cin, minus);

int num = -1;

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

if (books[i] == minus && available[i]) {

num = i;

break;

}

}

if (num != -1) {

cout << "You have successfully borrowed " << minus << ".\n";

cout << "\n";

available[num] = false;

taken.push\_back(minus);

} else {

cout << "Sorry, the book is unavailable.\n";

cout << "\n";

}

}

break;

case 3:

{

string plus;

cout << "Enter the book you want to return: ";

cin.ignore();

getline(cin, plus);

int num = -1;

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

if (taken[i] == plus) {

num = i;

break;

}

}

if (num != -1) {

cout << "You have successfully returned " << plus << ".\n";

cout << "\n";

taken.erase(taken.begin() + num);

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

if (books[i] == plus) {

available[i] = true;

break;

}

}

} else {

cout << "Sorry, the book was not borrowed.\n";

cout << "\n";

}

}

break;

case 4:

cout << "Borrowed books:\n";

for (const string& book : taken) {

cout << book << endl;

}

cout << "\n";

break;

case 5:

cout << "Goodbye!\n";

break;

default:

cout << "Invalid action. Please try again.\n";

cout << "\n";

}

} while (n != 5);

return 0;

}

Завдання №5: Практична# Програмування: Self Practice Task

#include <iostream>

int first(){

long a;

long int m;

long n = 0;

std::cin >> a;

long int l = a;

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

std::cin >> m;

n += m;

if (m <=0){

--l;

}

}

n -= l;

std::cout<<n;

return 0;

}

using namespace std;

int main (){

long a;

long int m;

long int n = 0;

cin >> a;

for (int i = 0; i < a && 1<=a<=100000; i++){

cin >> m;

if(0<=m<=1000000000){

if(m - 1 < 0){

n += 0;

} else{

n += m - 1;

}

} else{

return 1;

}

}

cout<<n<<endl;

cout<<"\n";

first();

return 0;

}

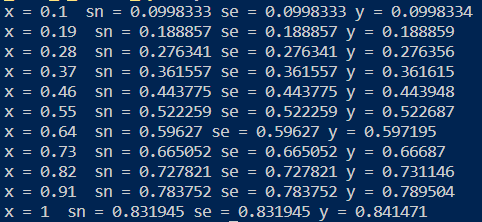
## **4. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1: Лаб# Програмування: VNS Lab 2 (Завдання 1)



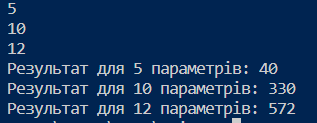
Затрачений час: 5 хв.

Завдання №2: Лаб# Програмування: VNS Lab 3 (Завдання 1)



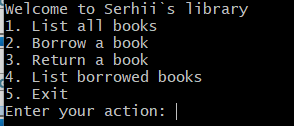
Затрачений час: 4 години.

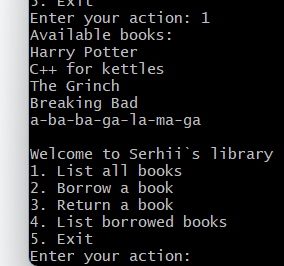
Завдання №3: Лаб# Програмування: VNS Lab 7 (Завдання 1)

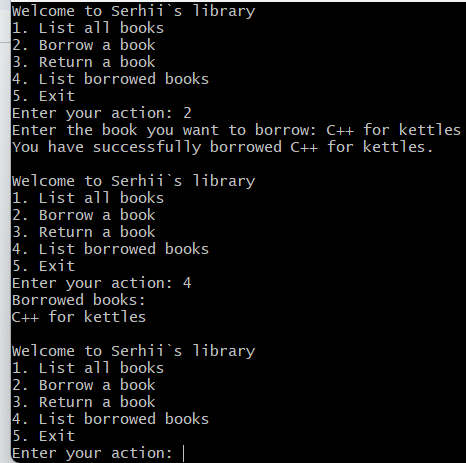


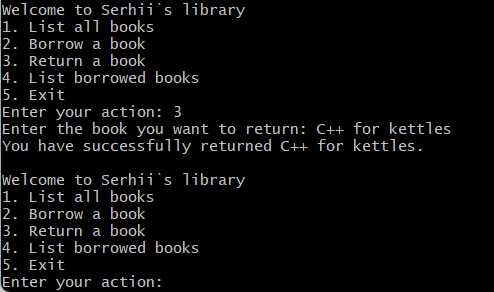
Затрачений час: 1 година(or less).

Завдання №4: Практична# Програмування: Class Practice Task



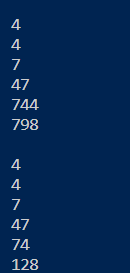






Затрачений час: 3 години.

Завдання №5: Практична# Програмування: Self Practice Task



Затрачений час: 30 хвилин.

# **Висновки:**

Отже, у ході роботи над епіком 3 я навчився працювати з різними типами циклів, зокрема while, do while, for i foreach, зрозумів різницю між ними та випадки, в яких їх доцільно застосовувати. Також я навчився працювати з функціями, ознайомився з поняттям параметрів та аргументів функцій, різними типами повернень(в тому числі void), зрозумів різницю між передачою аргументів функцій за значенням, посиланням та адресою. Окремо я ознайомився з поняттям еліпсису і як його використовувати для задання функцій з нефіксованою кількістю аргументів, а також ознайомився з рекурсивними функціями.

PR: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/556>