Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**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

Сокаль Олег Ігорович

# **Тема роботи:**

Знайомство з використанням циклів, функцій зі змінною кількістю параметрів та перевантаженням функцій. Використання функцій і їх рекурсій.

# **Мета роботи:**

Організація ітераційних і арифметичних циклів. Навчитись використовувати функції, перевантажені функції та рекурсію для створення програм, зокрема для обчислення функцій з використанням їхнього розкладу в степеневий ряд.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Оператори циклів.
* Тема №2: Оператори переходу.
* Тема №3: Функції.
* Тема №4: Перевантаження функцій.
* Тема №5: Функції зі змінною кількістю параметрів.
* Тема №6: Рекурсія.

1. Індивідуальний план опрацювання теорії:

* Тема №1 Оператори циклів.
  + Джерела Інформації:
    - Стаття.

https://www.programiz.com/cpp-programming/for-loop

https://www.geeksforgeeks.org/g-fact-40-foreach-in-c-and-java/

* + Що опрацьовано:
    - Ознайомився із операторами циклу for, while, do...while.
    - Також дізнався про використання оператора циклу foreach.
  + Статус: Ознайомлений.
  + Початок опрацювання теми: 30.10.2023
  + Звершення опрацювання теми: 30.10.2023
* Тема №2 Оператори переходу.
  + Джерела Інформації:
    - Стаття.

https://www.w3schools.com/cpp/cpp\_break.asp

* + Що опрацьовано:
    - Ознайомився із використанням операторів переходу break і continue.
  + Статус: Ознайомлений.
  + Початок опрацювання теми: 30.10.2023
  + Звершення опрацювання теми: 30.10.2023
* Тема №3: Функції.
  + Джерела Інформації:
    - Стаття.

https://www.w3schools.com/cpp/cpp\_functions.asp

https://www.w3schools.com/cpp/cpp\_function\_param.asp

* + Що опрацьовано:
    - Ознайомився із принципами оголошення і використання функцій.
    - Також дізнався про використання параметрів у функціях.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.11.2023
  + Звершення опрацювання теми: 15.11.2023
* Тема №4: Перевантаження функцій.
  + Джерела Інформації:
    - Стаття.

https://www.w3schools.com/cpp/cpp\_function\_overloading.asp

* + Що опрацьовано:
    - Ознайомився із використанням перевантажених функцій
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.11.2023
  + Звершення опрацювання теми: 15.11.2023
* Тема №5Функції зі змінною кількістю параметрів.
  + Джерела Інформації:
    - Відео.

https://www.youtube.com/watch?v=oDC208zvsdg

* + - Стаття.

https://en.cppreference.com/w/c/variadic

https://www.geeksforgeeks.org/variadic-functions-in-c/

* + Що опрацьовано:
    - Ознайомився із принципами роботи з функціями зі змінною кількістю параметрів.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.11.2023
  + Звершення опрацювання теми: 15.11.2023
* Тема №6 Рекурсія.
  + Джерела Інформації:
    - Стаття.

https://www.w3schools.com/cpp/cpp\_functions\_recursion.asp

* + Що опрацьовано:
    - Дізнався про застосування рекурсійних функцій.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.11.2023
  + Звершення опрацювання теми: 15.11.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 2

* Варіант завдання 19
* Деталі завдання:

Знайти суму зазначених елементів.

* Важливі деталі для врахування в імплементації програми:

Потрібно використати оператор циклу do\_while.

Завдання №2 VNS Lab 3

* Варіант завдання 19

Для х, що змінюється від a до b з кроком (b-a)/k, де (k=10), обчислити

функцію f(x), використовуючи її розклад в степеневий ряд у двох випадках:

а) для заданого n = 20;

б) для заданої точності ε (ε=0.0001).

* Важливі деталі для врахування в імплементації програми

Для порівняння знайти точне значення функції.

* Математична модель:

Завдання №3 VNS Lab 7

* Варіант завдання 19
* Деталі завдання

1. Написати функцію belong зі

змінною кількістю параметрів, що визначає скільки точок з координатами

(х,у) належать заданому кругу.

Написати викликаючу функцію main, що

звертається до функції belong не менше трьох разів з кількістю параметрів 3,

9, 11.

2. Написати перевантажені функції для додавання дійсних чисел і додавання комплексних чисел й основну програму, що їх викликає.

а) для масиву цілих чисел знаходить мінімальний елемент;

б) для рядка знаходить довжину найкоротшого слова .

* Важливі деталі для врахування в імплементації програми

Потрібно використати перевантаження функції.

Завдання №4 Class Practice Work

* Деталі Завдання

Cтворити просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

* Важливі деталі для врахування в імплементації програми

Програма повинна вміти перерахувати всі книги, дозволити взяти книгу (за наявності), дозволити повернення книги.

Завдання №5 Self Practice Work

* Деталі завдання

1.Вам дано 2 масиви розміром N та M. Значення у цих масивах унікальні.

Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

2. У світі Атод сестри Ліна і Рілай люблять грати у гру. У них є дошка із 8-ми рядків і 8-ми стовпців. На перетині

i-го рядкa і

j-го стовпця лежить магічна куля, яка може світитись магічним світлом (тобто у них є 64 кулі). На початку гри деякі кулі світяться, а деякі ні... Далі вони обирають

N куль і для кожної читають магічне заклиння, після чого всі кулі, які лежать на перетині стовпця і рядка обраної кулі міняють свій стан (ті що світяться - гаснуть, ті, що не світяться - загораються).

Також вони вирішили трохи Вам допомогти і придумали спосіб як записати стан дошки одним числом

a

із 8-ми байт, а саме (див. Примітки):

* Молодший байт задає перший рядок матриці;
* Молодший біт задає перший стовпець рядку;
* Значення біту каже світиться куля чи ні (0 - ні, 1 - так);

Тепер їх цікавить яким буде стан дошки після виконання

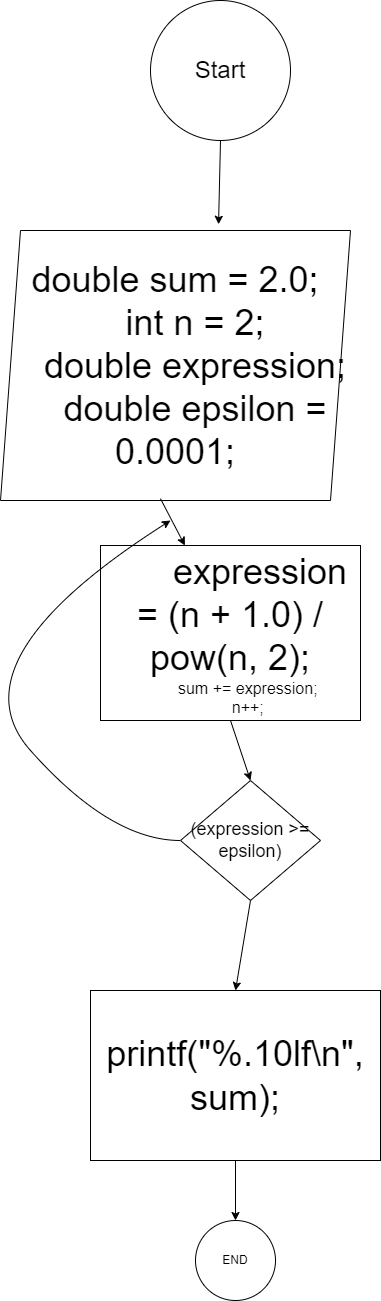
N заклинань і вони дуже просять Вас їм допомогти.

3.Ви з’явились у світі під назвою Атод посеред Пустелі Безправ’я. Так сталось, що Ви попали саме в той час і місце, де ведеться битва між чаклункою Ліною і темними силами, які хочуть знищити цей світ. На жаль, трапилась халепа, бо деякі слова із книги чар були пошкоджені під час битви. Одне таке слово можна відновити виконавши ритуал зцілення над пошкодженими буквами. Ритуал зцілення можна виконати на всіх **підряд** розташованих **пошкоджених** буквах. Вам не залишається нічого іншого як допомогти Ліні відновити ці слова і сказати скільки мінімально треба провести таких ритуалів, щоб прочитати одне з наймогутніших у цьому світі заклять - Поневолення Дракона!

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

Програма №1 VNS Lab 2

* Блок-схема:

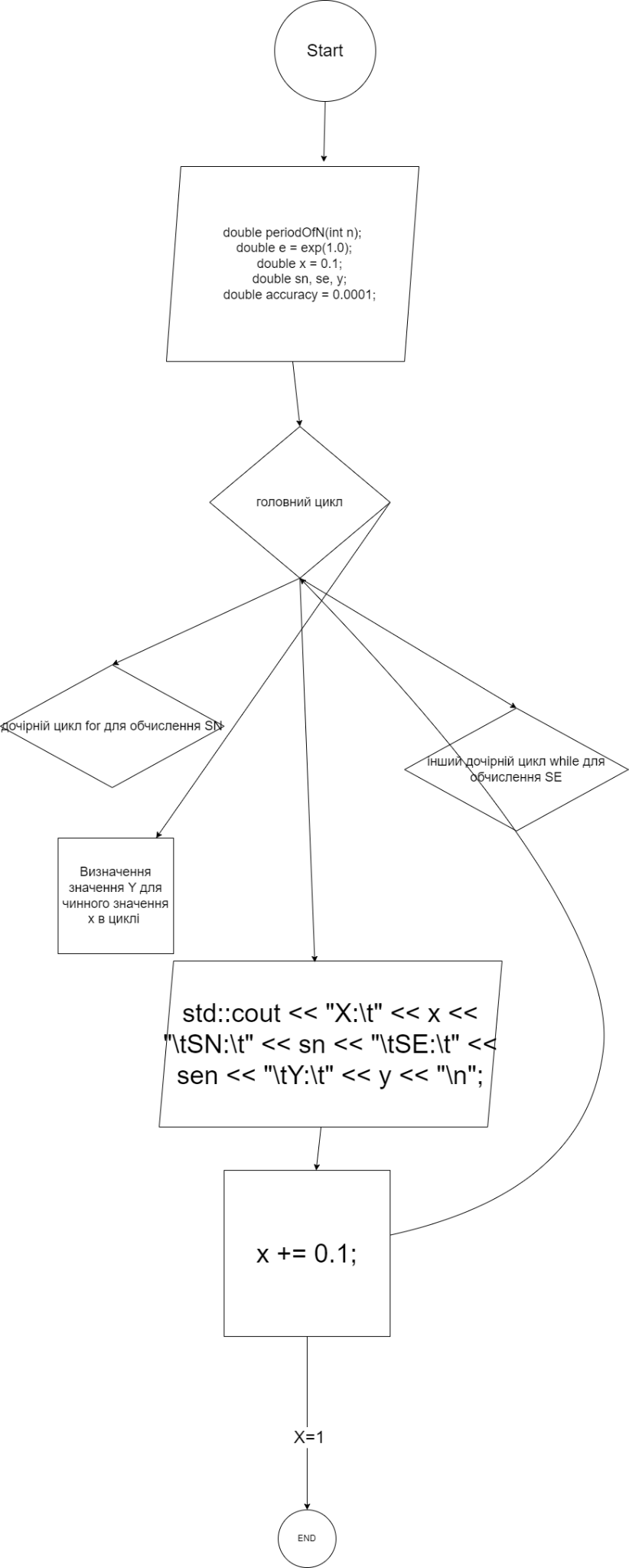


*Рисунок 1: Блок-схема до програми №1*

* Планований час на реалізацію 1 день.

Програма №2 VNS Lab 3

* Блок-схема



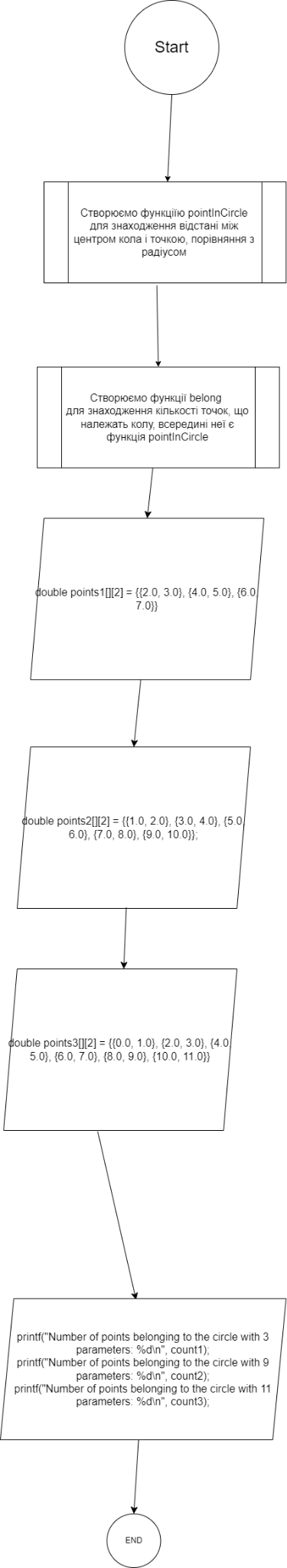
*Рисунок 2: Блок-схема до програми №2*

* Планований час на реалізацію 1 день.
* Важливі деталі для врахування в імплементації

Алгоритм розв’язання завдання зводиться до трьох циклів, причому два з них вкладені в третій. Внутрішні цикли підсумують доданки при фіксованому параметрі x, один (арифметичний для заданого n), інший (ітераційний для заданої точності ε. При організації цих циклів варто звернути увагу на правильний вибір формули для обчислення елемента ряду a(n) і правильне присвоєння початкових значень змінним циклу. Зовнішній цикл організує зміну параметра х

Програма №3 VNS Lab 7

* Блок-схема



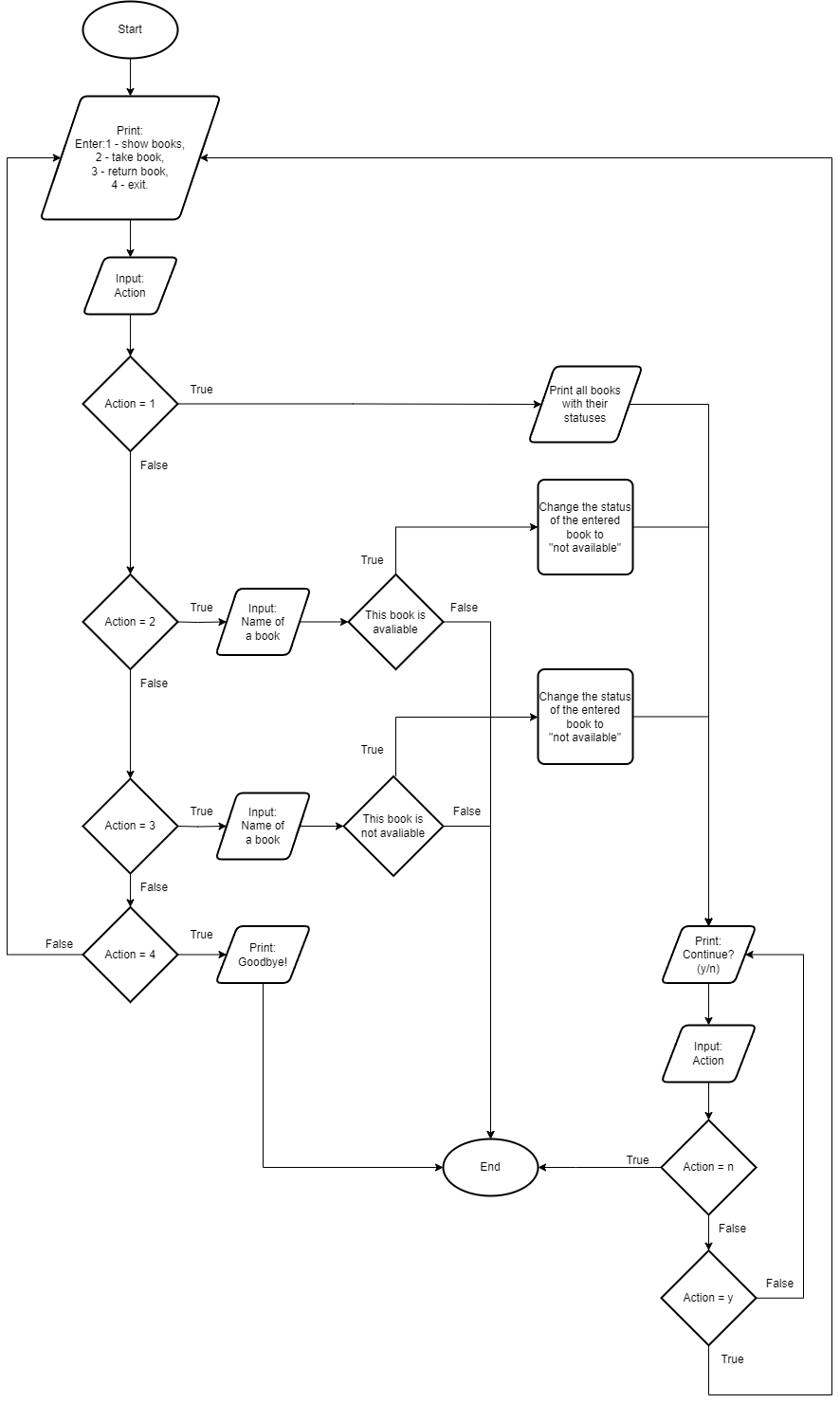
*Рисунок 3: Блок-схема до програми №3*

* Планований час на реалізацію 1 день.
* Важливі деталі для врахування в імплементації

Використовувати тип даних float.

Програма №4 Class Practice Work

* Блок-схема



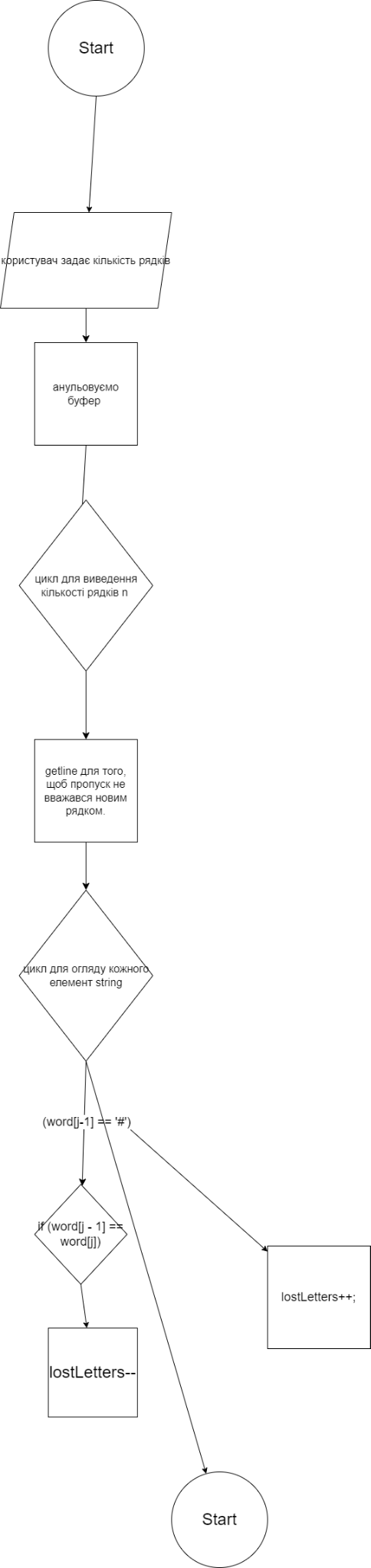
*Рисунок 4: Блок-схема до програми №4*

* Планований час на реалізацію 1 день.
* Важливі деталі для врахування в імплементації

Використати масив або вектор для зберігання назв книг та інший масив або вектор для збереження стану доступності кожної книги.

Програма №5 Self Practice Work

* Блок-схема



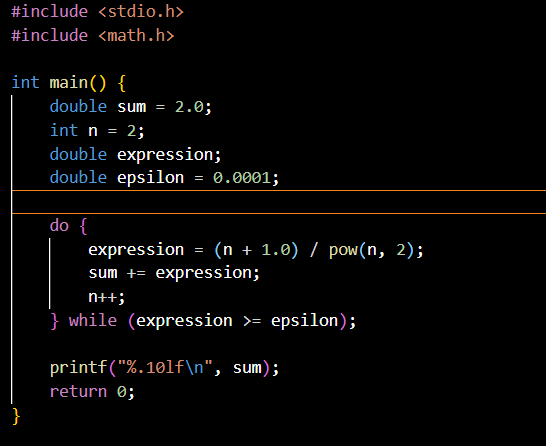
*Рисунок 5: Блок-схема до програми №5*

* Планований час на реалізацію 1 день.
* Важливі деталі для врахування в імплементації:
* всі дані менше 100

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

* Додаткова конфігурація середовища не потрібна.

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

Завдання №1 VNS Lab 2  


*Рисунок 6: Код до програми №1*

Програма знаходить суму цілих додатних парних чисел менших епсилона, використовуючи цикл for. Отриманий результат програма виводить у консоль.

Завдання №2 VNS Lab 3  
#include <iostream>

#include <cmath>

double periodOfN(int n) {

if (n == 0) {

return 1;

} else {

return n \* periodOfN(n - 1);

}

}

int main() {

double e = exp(1.0);

double x = 0.1;

double sn, se, y;

double accuracy = 0.0001;

for (double i = x; i <= 1; i += 0.1) {

sn = 1;

double sen = 0;

for (int n = 1; n <= 20; n++) {

sn += pow(2 \* i, n) / periodOfN(n);

}

int n = 0;

while (accuracy < (pow(2 \* i, n) / periodOfN(n))) {

se = 0.0;

se = pow(2 \* i, n) / periodOfN(n);

sen +=se;

n++;

}

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

std::cout << "X:\t" << x << "\tSN:\t" << sn << "\tSE:\t" << sen << "\tY:\t" << y << "\n";

x += 0.1;

}

return 0;

}

*Рисунок 7: Код до програми №2*

Спочатку програма знаходить значення Х для яких буде обчислювати функцію. Далі, використовуючи розклад функції в степеневий ряд, програма обчислює значення функції для заданого n та для заданої точності ε. Потім виводить отримані результати в консоль.

Завдання №3 VNS Lab 7  
#include <stdio.h>

#include <math.h>

int pointInCircle(double x, double y, double R) {

double distance = sqrt(x \* x + y \* y);

return distance <= R;

}

int belong(int R, int numPoints, double points[][2]) {

int count = 0;

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

double x = points[i][0];

double y = points[i][1];

if (pointInCircle(x, y, R)) {

count++;

}

}

return count;

}

int main() {

int R = 100;

double points1[][2] = {{2.0, 3.0}, {4.0, 5.0}, {6.0, 7.0}};

int count1 = belong(R, sizeof(points1) / sizeof(points1[0]), points1);

printf("Number of points belonging to the circle with 3 parameters: %d\n", count1);

double points2[][2] = {{1.0, 2.0}, {3.0, 4.0}, {5.0, 6.0}, {7.0, 8.0}, {9.0, 10.0}};

int count2 = belong(R, sizeof(points2) / sizeof(points2[0]), points2);

printf("Number of points belonging to the circle with 9 parameters: %d\n", count2);

double points3[][2] = {{0.0, 1.0}, {2.0, 3.0}, {4.0, 5.0}, {6.0, 7.0}, {8.0, 9.0}, {10.0, 11.0}};

int count3 = belong(R, sizeof(points3) / sizeof(points3[0]), points3);

printf("Number of points belonging to the circle with 11 parameters: %d\n", count3);

return 0;

}

—-------------------------

#include <iostream>

#include <string>

using namespace std;

int min\_element(int n, int a[]) {

int min = a[0];

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

if (a[i] < min)

min = a[i];

return min;

}

int findMinWordLength(const int arr[], int n) {

int min = arr[0];

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

if (min > arr[i]) {

min = arr[i];

}

}

return min;

}

int main() {

int var;

cout << "яке завдання:(1,2)";

cin >> var;

if (var == 2) {

int n = 0;

std::string sentence;

cin.ignore();

getline(std::cin, sentence);

int count = 0;

int arr[100] = {0};

for (int i = 0; i < sentence.length(); i++) {

if (sentence[i] != ' ' && sentence[i] != '\0') {

count++;

} else {

arr[n] = count;

count = 0;

n++;

}

}

int min = findMinWordLength(arr, n);

std::cout << "Length of smallest word: " << min << std::endl;

return 0;

}

if (var == 1) {

int a[100];

int n;

cout << "введіть кількість чисел: ";

cin >> n;

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

cout << "введіть число " << i + 1 << ": ";

cin >> a[i];

}

int result = min\_element(n, a);

cout << "Мінімальний елемент: " << result << endl;

return 0;

}

return 0;

}

*Рисунок 8: Код до програми №3*

Програма знаходить добуток 3, 7, 11 дійсних чисел. Потім суму дійсних і комплексних чисел. Отримані результати програма виводить у консоль.

Завдання №4 Class Practice Work  
#include <iostream>

#include <string>

using namespace std;

int main() {

int count=0;

string answer;

int max = 5;

string books[5] = {"1984", "Woman in Berlin", "Physchology of influence", "Financier", "Animal Farm A Fairy Story"};

bool available[5] = {true, true, true, true, true};

int pick;

char choice;

string m;

using namespace std;

LabelMenu:

if(count==0){

m="Star?(y/n): ";

}

else{

m= "Keep going?(y|n): ";

}

cout << "" << m;

cin >> answer;

while (answer == "y") {

count++;

cout << " Main Menu\n";

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

cout << "2. Take a one\n";

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

cout << "Choose the option(1/2/3/4): ";

cin >> choice;

if (choice == '1') {

cout << "List of books:\n";

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

cout << i + 1 << ". " << books[i] << " (" << (available[i] ? "Доступна" : "Позичена") << ")\n";

}

goto LabelMenu;

} else if (choice == '2') {

cout << "What one do you want to pick (1-5): ";

cin >> pick;

if (pick >= 1 && pick <= max) {

if (available[pick - 1]==true) {

cout << "You took a book '" << books[pick - 1] << "'.\n";

available[pick - 1] = false;

goto LabelMenu;

}

else {

cout << "This book was already taken\n";

goto LabelMenu;

}

} else {

cout << "Unknown number\n";

goto LabelMenu;

}

} else if (choice == '3') {

cout << "What one do you want to return? (1-5): ";

cin >> pick;

if (pick >= 1 && pick <= max) {

if (available[pick - 1] == false) {

cout << "You've gave <<" << books[pick - 1] << ">> back\n";

available[pick - 1] = true;

} else {

cout << "You haven't taken it from the library\n";

goto LabelMenu;

}

} else {

cout << "it's no available book with this number.\n";

goto LabelMenu;

}

}

else {

cout << "Maybe mistake. Try again\n";

goto LabelMenu;

}

}

return 0;

}

*Рисунок 9: Код до програми №4*

Програма отримує команду від користувача. Залежно від введеної команди програма показує весь каталог книжок в бібліотеці із їхніми статусами, дозволяє користувачу взяти, або повернути книгу, або вийти з програми. Після кожної дії користувача запитують, чи потрібно продовжувати роботу програми.  
  
Завдання №5 Self Practice Work  
#include <iostream>

#include <string>

using namespace std;

int main() {int n;

int lostLetters = 0;

cin >> n;

cin.ignore();

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

string word;

getline(cin, word);

for (int j = 1; j <= word.length(); j++) {

if (word[j-1] == '#') {

if (word[j - 1] == word[j]) {

lostLetters--;

}

lostLetters++;

}

}

}

cout << lostLetters << endl;

return 0;

}

—---------------------

#include <iostream>

using namespace std;

int main() {

int arr[100] = {0}, arr1[100] = {0};

int n, n1;

int unique = 0;

int similar = 0;

cin >> n;

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

cin >> arr[i];

}

cin >> n1;

for (int i = 0; i < n1; i++) {

cin >> arr1[i];

}

for (int i = 0; i < n1; i++) {

int found = 0;

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

if (arr1[i] == arr[j]) {

similar++;

}

}

}

unique= (n1+n) - similar;

cout << similar << endl;

cout << unique << endl;

return 0;

}

*Рисунок 10: Код до програми №5*

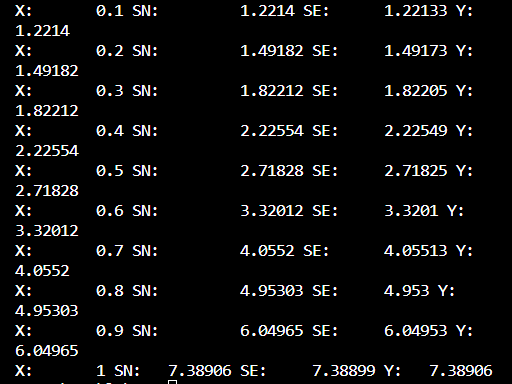
## 1.Програма знаходить кількість символів #, але декілька послідовних теж рахує за один.

## 2. Знаходить кількість однакових та різних символів у двох різних масивах.

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

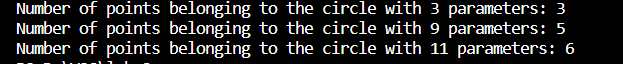
Завдання №1  

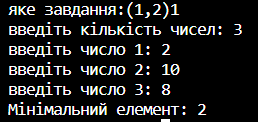

*Рисунок 11: Приклад виконання програми №1*Час затрачений на виконання завдання 1 день.

Завдання №2  
**

*Рисунок 12: Приклад виконання програми №2*

Час затрачений на виконання завдання1 день.

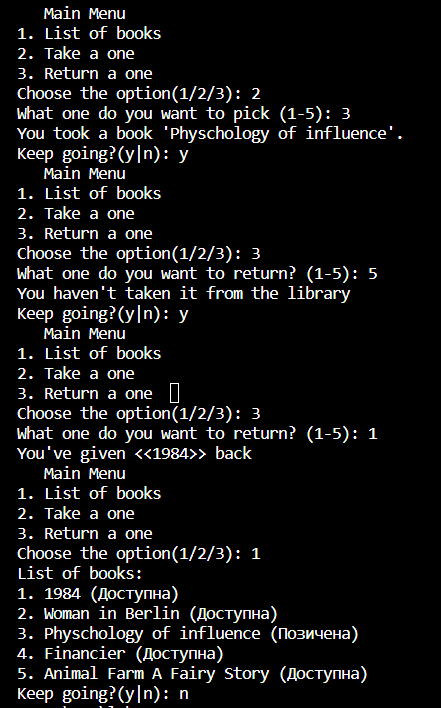
Завдання №3 **

**

**

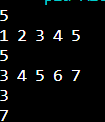
*Рисунок 13: Приклад виконання програми №3*

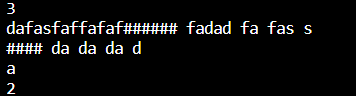
Час затрачений на виконання завдання 1 день.

Завдання №4 **

*Рисунок 14: Приклад виконання програми №4*

Час затрачений на виконання завдання 1 день.

Завдання №5 **

**

*Рисунок 15: Приклад виконання програми №5*

**

**

*Рисунок 16: Завдання №5 зараховане в системі Алготестер*

Час затрачений на виконання завдання 1 день.

# [**https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/486/files#diff-6b745ab2a27b5766500308592c1545ef3580d26eb2adb735254493926bb87bd3**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/486/files#diff-6b745ab2a27b5766500308592c1545ef3580d26eb2adb735254493926bb87bd3)

# **Висновки:**

Під час роботи над цим розділом я ознайомився із принципами роботи із функціями, перевантаженими функціями, рекурсією. Створив низку програм, що використовують дані принципи та передбачають організацію ітераційних і арифметичних алгоритмів.