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-11

Винницький Андрій Олегович

# **Тема роботи:**

Цикли та вкладені цикли. Функції. Перевантаження функцій. Рекурсія

# **Мета роботи:**

Навчитися працювати з вкладеними циклами, такими як for, while і do while. Написати завдання з VNS і практичні завдання. Також детальніше познайомитися з функціями, навчитися їх перевантажувати. Попрацювати з рекурсією. Всі зроблені задачі і звіт запушити на GitHub і зробити pull request.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли та вкладені цикли.
* Тема №2: Перевантаження функцій. Рекурсія.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли та вкладені цикли.
  + Джерела Інформації
    - <https://www.geeksforgeeks.org/cpp-loops/>
  + Що опрацьовано:
    - Опрацьовано цикли for, while і do while, а також вкладені цикли.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 02.11.2023
  + Звершення опрацювання теми: 08.11.2023
* Тема №2: Перевантаження функцій. Рекурсія.
  + Джерела Інформації:
    - <https://www.geeksforgeeks.org/function-overloading-vs-function-overriding-in-cpp/>
    - <https://www.geeksforgeeks.org/function-overloading-c/>
    - <https://www.geeksforgeeks.org/function-overloading-and-return-type-in-cpp/>
    - <https://www.geeksforgeeks.org/cpp-recursion/>
    - <https://www.programiz.com/cpp-programming/recursion>
    - <https://learn.microsoft.com/ru-ru/cpp/c-runtime-library/reference/va-arg-va-copy-va-end-va-start?view=msvc-170>
  + Що опрацьовано:
    - Ознайомився з рекурсією і перевантаженням функцій, отримані знання застосував на практиці.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 10.11.2023
  + Звершення опрацювання теми: 15.11.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 2 - Task 1

* Варіант завдання 9
* Порахувати суму n перших елементів ряду

Завдання №2 VNS Lab 3 - Task 1

* Варіант завдання 9
* Порахувати значення функції для різних x, розклавши її в ряд Тейлора. Зробити це за кількістю елементів і за точністю в 0,0001.

Завдання №3 VNS Lab 7 - Task 1-1

* Варіант завдання 9
* Написати функцію max зі змінною кількістю параметрів, що визначає мінімальне значення серед типу double або int. Написати викликаючу функцію main, що звертається до функції min не менше трьох разів з кількістю параметрів 5, 10, 12
* Другою частиною завдання було написати перевантажену функцію для додавання дробів.

Завдання №4 Class Practice Work

* Написати просту програму яка буде обслуговувати бібліотеку. Програма повинна вміти перелічити всі доступні книжки, дозволити взяти чи повернути книжку за наявності, а також вийти з бібліотеки.

Завдання №5 Self Practice Work

* Реалізувати алгоритм merge sort

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

Програма №1 VNS Lab 2 - Task 1

* Блок-схема

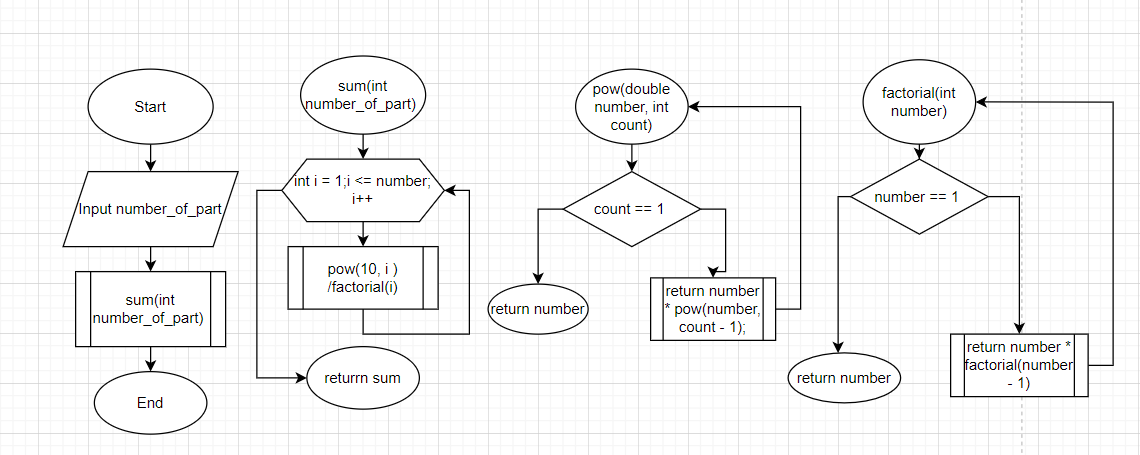


Рисунок 1: Блок-схема до програми 1

* Планований час на реалізацію 15 хв

Програма №2 VNS Lab 3 - Task 1

* Блок-схема

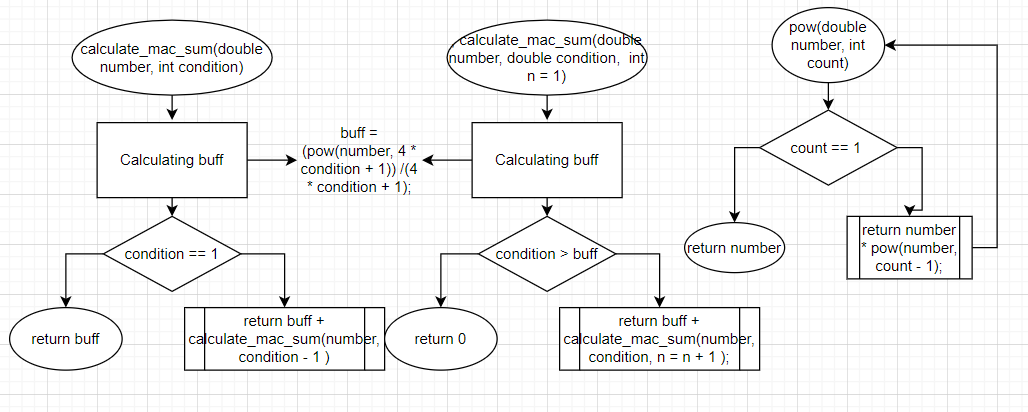


Рисунок 2:Блок-схема до програми 2

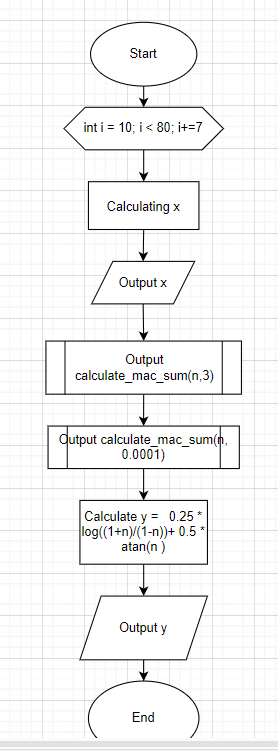


Рисунок 3:Блок-схема до програми 2

* Планований час на реалізацію 15 хв

Програма №3 VNS Lab 7 - Task 1

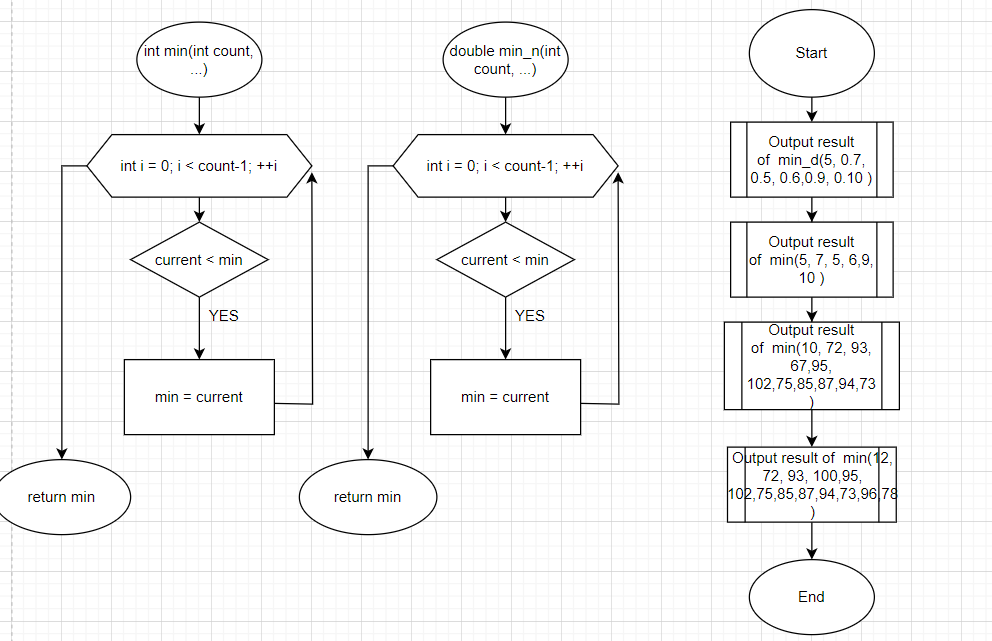
* Блок-схема
* 

Рисунок 4:Блок-схема до програми 3

* Планований час на реалізацію 30хв

Програма №3 VNS Lab 7 - Task 2

* Блок-схема

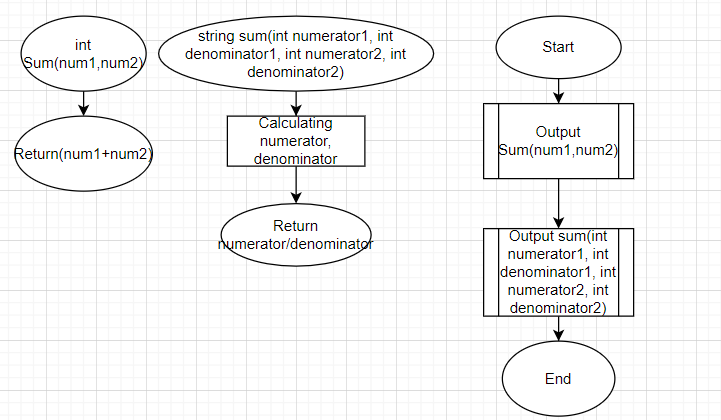
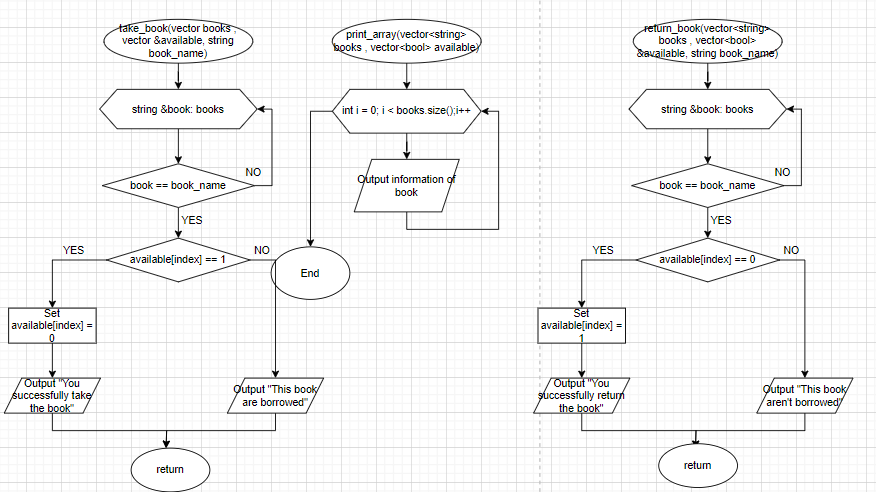


Рисунок 5:Блок-схема до програми 4

Програма №4 Class Practice Work

* Блок-схема



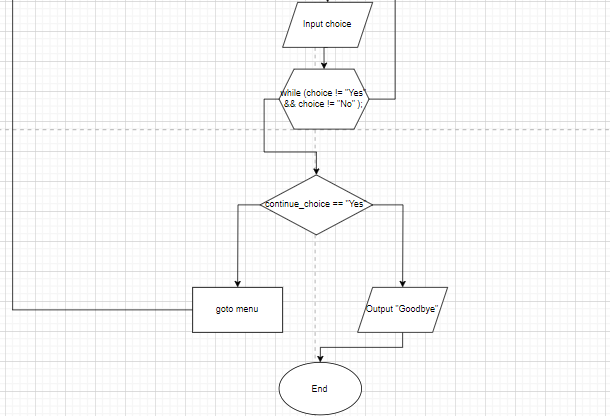
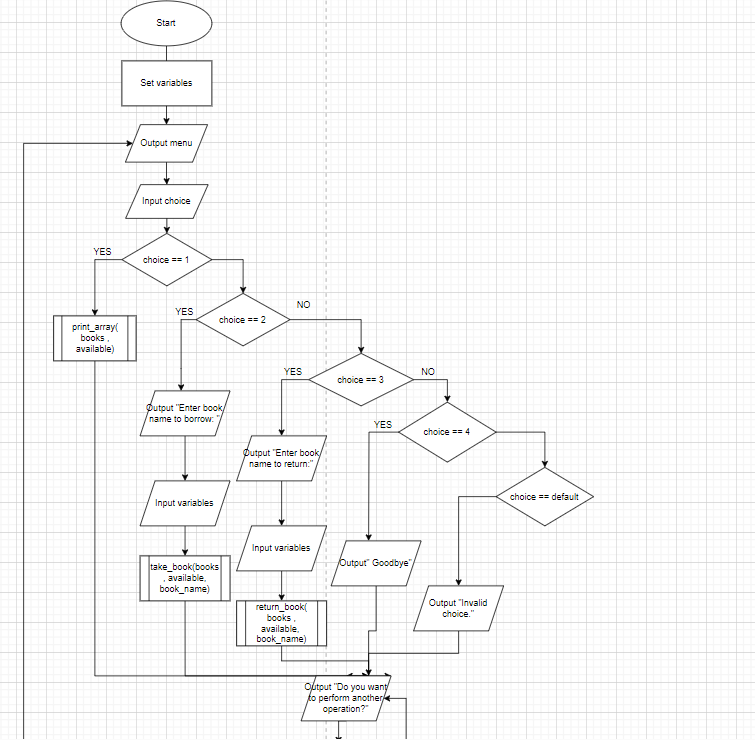


Рисунок : Блок-схема до програми 5

* Планований час на реалізацію 1.5 год

Програма №5 Self Practice Work

Планований час на реалізацію 20 хв

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

Завдання №1 VNS Lab 2 - Task 1

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/648/files#diff-172f8092b18b52ecc09a0419b5653fc49025cd4c2b3581f0c89c947b6ab6581a>

#include<iostream>  
  
using namespace std;  
double pow(double number, int count)  
{  
  
 if(count == 1 ){  
 return number;  
 }  
 return number \* pow(number, count - 1);  
  
}  
int factorial(int number)  
{  
 if(number == 1)  
 {  
 return number;  
 }  
 return number \* factorial(number - 1);  
  
}  
  
double sum(int number)  
{  
 double sum = 0.0;  
 for(int i = 1;i <= number; i++)  
 {  
 sum += pow(10, i ) /factorial(i);  
 }  
 return sum;  
}  
  
  
int main()  
{  
 int number\_of\_part;  
 cin >> number\_of\_part;  
 printf ("Result: %.4f%" , sum(number\_of\_part));  
 return 0;  
}

Час витрачений : 10 хв.

Завдання №2 VNS Lab 3 - Task 1

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/648/files#diff-ba0c2dc857db34347400099b94b94501a01f2fe5707a3d25fc60979a576c2f1e>

#include <iostream>  
#include<cmath>  
using namespace std;  
  
  
double pow(double number, int condition)  
{  
 if (condition == 1)  
 {  
 return number;  
 }  
 return number \* pow(number, condition - 1);  
}  
  
  
double calculate\_mac\_sum(double number, int condition)  
{  
  
 double buff = (pow(number, 4 \* condition + 1)) /(4 \* condition + 1);  
  
 if(condition == 1)  
 {  
 return buff;  
 }  
 return buff + calculate\_mac\_sum(number, condition - 1 );  
  
}  
double calculate\_mac\_sum(double number, double condition, int n = 1)  
{  
 double buff = (pow(number, 4 \* n + 1)) /(4 \* n + 1);  
  
 if(condition > buff)  
 {  
 return 0;  
 }  
 return buff + calculate\_mac\_sum(number, condition, n = n + 1 );  
}  
  
int main() {  
for (int i = 10; i < 80; i+=7)  
{  
 double n = i /100.00;  
 cout<< "Argument X: "<< n<<endl;  
 cout<< "Tailor formula by number monomials: "<<calculate\_mac\_sum(n,3)<< endl;  
 cout<< "Tailor formula by accuracity: "<< calculate\_mac\_sum(n, 0.0001)<< endl;  
 cout<< "Y= "<< 0.25 \* log((1+n)/(1-n))+ 0.5 \* atan(n )<<endl;  
  
}  
return 0;  
}

Час витрачений : 20 хв.

Завдання №3 VNS Lab 7 - Task 1

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/648/files#diff-01258a325ab147c65bc383ed44119935910295a70fbd86d226d6abdbd537eef0>

#include <iostream>  
  
#include <stdarg.h>  
using namespace std;  
  
int min(int count, ...) {  
 va\_list list;  
 va\_start(list, count);  
 int min = va\_arg(list, int);  
 for (int i = 0; i < count-1; ++i) {  
 int current = va\_arg(list, int);  
 if (current < min){  
 min = current;  
 }  
 }  
  
 va\_end(list);  
 return min;  
}  
  
double min\_d(int count, ...) {  
 va\_list list;  
 va\_start(list, count);  
 double min = va\_arg(list, int);  
 for (int i = 0; i < count-1; ++i) {  
 double current = va\_arg(list, double);  
 if (current < min){  
 min = current;  
 }  
 }  
  
 va\_end(list);  
 return min;  
}  
  
  
  
int main ()  
{  
 cout <<"Five: "<< min\_d(5, 0.7, 0.5, 0.6,0.9, 0.10 )<<endl;  
 cout <<"Five: "<< min(5, 7, 5, 6,9, 10 )<<endl;  
 cout <<"Ten: "<< min(10, 72, 93, 67,95, 102,75,85,87,94,73 )<<endl;  
 cout <<"Twelve: "<< min(12, 72, 93, 100,95, 102,75,85,87,94,73,96,78 )<<endl;  
 return 0;  
}

Час витрачений : 30 хв.

Завдання №3 VNS Lab 7 - Task 2

Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/648/files#diff-c761e9f372c63bb1b9575a6f41a305f33846f9b4474d27b83955dbbc2f4d8ca0>

#include <iostream>  
using namespace std;  
  
double sum(double num1, double num2)  
{  
 return num1 + num2;  
}  
  
string sum(int numerator1, int denominator1, int numerator2, int denominator2)  
{  
 int denominator = denominator1\*denominator2;  
 int numerator = numerator1\*denominator2 +numerator2\*denominator1;  
 return to\_string(numerator) +'/' + to\_string(denominator);  
}  
  
int main()  
{  
 cout<< "Decimal fraction: "<< sum(0.73, 0.96)<<endl;  
 cout<< "Basic fraction: "<< sum(7, 13, 8,23);  
 return 0;  
  
}

Час витрачений : 10 хв.

Завдання №4 Class Practice Work

Посилання :<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/648/files#diff-2eceee3dfdf6d97d6410f95a3f0d70ac31e7fcc81e865664c53c2734498e6f35>

#include <iostream>  
#include <vector>  
  
using namespace std;  
  
void print\_array(vector<string> books , vector<bool> available);  
void take\_book(vector<string> books , vector<bool> &available , string book\_name);  
void return\_book(vector<string> books , vector<bool> &available, string book\_name);  
  
int main() {  
 vector<string> books = **{**"Harry Potter", "The Hobbit", "Pride and Prejudice", "1984"**}**;  
 vector<bool> available = **{**1, 1, 1, 1**}**;  
 int choice;  
 string bookName;  
  
  
 menu:  
 cout << "Library Management\n"  
 "1. List all books\n"  
 "2. Borrow a book\n"  
 "3. Return a book\n"  
 "4. Exit\n"  
 "Enter your choice: ";  
 cin >> choice;  
  
  
 switch(choice) {  
 case 1:  
 print\_array(books, available);  
 break;  
 case 2:  
 cin.ignore();  
 cout << "Enter book name to borrow: ";  
 getline(cin, bookName);  
  
  
 take\_book(books, available, bookName);  
 break;  
 case 3:  
 cin.ignore();  
 cout << "Enter book name to return: ";  
 getline(cin, bookName);  
  
 return\_book(books, available, bookName);  
  
 break;  
 case 4:  
 cout << "Goodbye!";  
 break;  
 default:  
 cout << "Invalid choice.\n";  
 goto menu;  
 }  
  
  
 string continue\_choice;  
 do {  
 cout << "Do you want to perform another operation?\n - Yes\n - No: ";  
 cin >> continue\_choice;  
 } while (continue\_choice != "Yes" && continue\_choice != "No" );  
  
  
 if (continue\_choice == "Yes") {  
 goto menu;  
 } else {  
 cout << "Goodbye!";  
 }  
  
  
 return 0;  
}  
  
void print\_array(vector<string> books , vector<bool> available)  
{  
  
 for (int i = 0; i < books.size();i++)  
 {  
 cout<< books[i]<< ": "<< (available[i] ? "Available" : "Borrowed") << "\n";  
 }  
}  
void take\_book(vector<string> books , vector<bool> &available, string book\_name)  
{  
 for (string &book: books)  
 {  
 if (book == book\_name)  
 {  
 int index = &book - &books[0];  
 if (available[index])  
 {  
 available[index] = 0;  
 cout << "You successfully take the book\n";  
 return;  
 }  
 else  
 {  
 cout<<"This book are borrowed\n";  
 return;  
 }  
 }  
 }  
}  
void return\_book(vector<string> books , vector<bool> &available, string book\_name)  
{  
 for (string &book: books)  
 {  
 if (book == book\_name)  
 {  
 int index = &book - &books[0];  
 if (!available[index])  
 {  
 available[index] = 1;  
 cout << "You successfully return the book\n";  
 return;  
 }  
 else  
 {  
 cout<<"This book aren't borrowed\n";  
 return;  
 }  
 }  
 }  
}

Час витрачений : 40 хв.

Завдання №5 Self Practice Work.

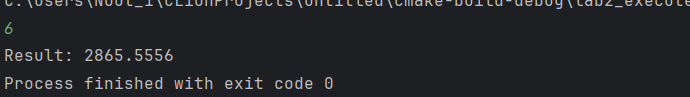
Посилання: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/648/files#diff-77115d7690c1412bc040f91e818c97d7b4993e48a778a8e8ee38ec30535d5592>

#include <iostream>  
#include <vector>  
#include <ctime>  
using namespace std;  
  
void print\_array(vector<int> num, int start , int end)  
{  
 for(int i = start; i < end; i++)  
 {  
 cout << num[i]<<" ";  
 }  
}  
  
void merging(vector<int>& num\_arr, int start ,int midle, int end)  
{  
 vector<int> buff\_right(num\_arr.begin() + midle + 1, num\_arr.begin() + end + 1);  
 vector<int> buff\_left(num\_arr.begin() + start, num\_arr.begin() + midle + 1);  
 int ind = 0;  
 while (!buff\_left.empty() && !buff\_right.empty())  
 {  
 if (buff\_left[ind] > buff\_right[ind])  
 {  
 num\_arr[start] = buff\_right[ind];  
 buff\_right.erase(buff\_right.begin() + ind);  
 start++;  
 }  
 else  
 {  
 num\_arr[start] = buff\_left[ind];  
 buff\_left.erase(buff\_left.begin() + ind);  
 start++;  
 }  
  
 }  
 if(buff\_left.empty())  
 {  
 for(int i : buff\_right)  
 {  
 num\_arr[start] = i;  
 start++;  
 }  
  
 }  
 else if(buff\_right.empty())  
 {  
 for(int i : buff\_left)  
 {  
 num\_arr[start] = i;  
 start++;  
 }  
  
 }  
 }  
  
void merge\_sort(vector<int>& num\_arr, int start , int end)  
{  
 if (end != start)  
 {  
 int mid = (end + start)/2;  
  
 merge\_sort(num\_arr, start , mid);  
 merge\_sort(num\_arr, mid + 1 ,end);  
  
 merging(num\_arr, start, mid, end);  
 }  
  
}  
  
int main()  
{  
 srand(time(0));  
 vector<int> nums;  
 for(int i = 0; i < 100;i++)  
 {  
 nums.push\_back(rand() % 100);  
 }  
  
 merge\_sort(nums, 0 , nums.size() - 1);  
 cout<<"Sorted array: ";  
 print\_array(nums, 0, nums.size());  
 return 0;  
}

Час витрачений : 55 хв.

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

Завдання №1 VNS Lab 2 - Task 1



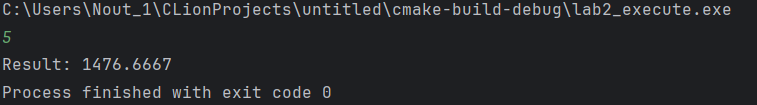


Рисунок : Код обчислює суму ряду довжиною в n за певною формулою

Завдання №2 VNS Lab 3 - Task 1

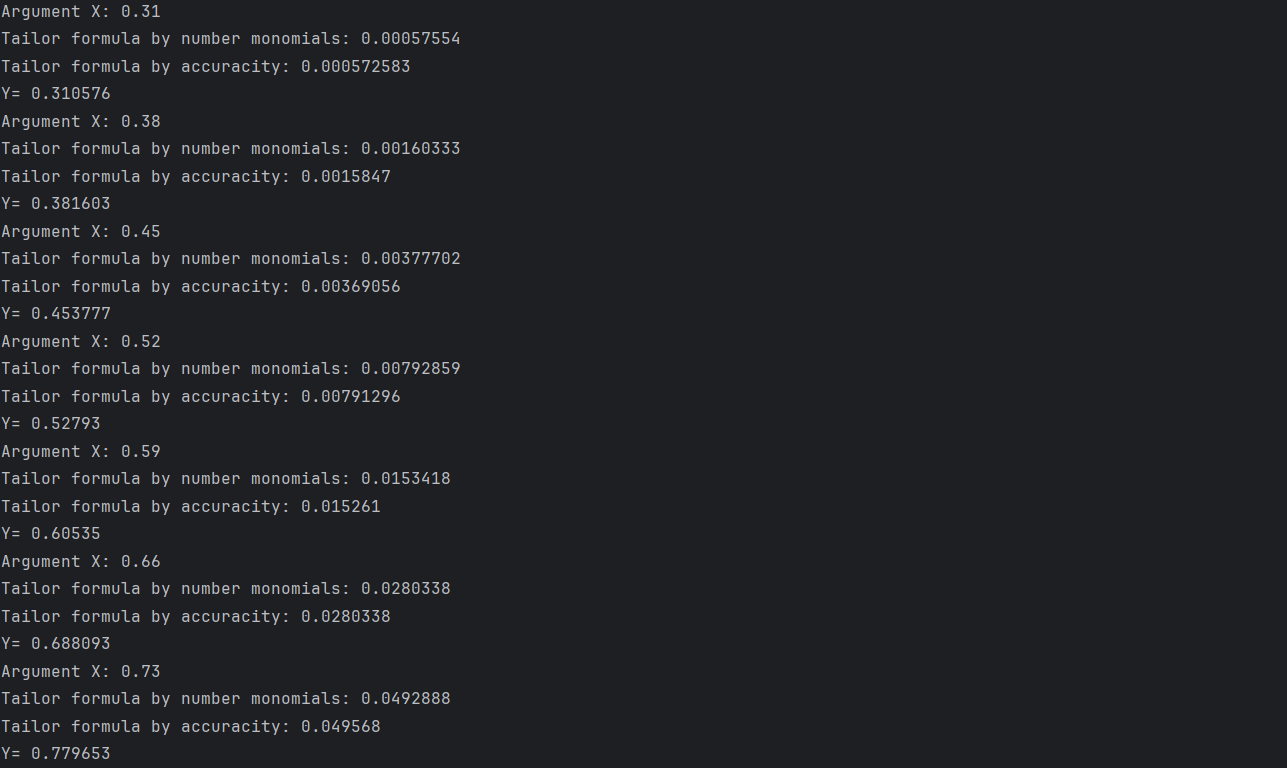


Рисунок :Програма рахує значення функції для різних x, розкладає її в ряд Тейлора. Робить це за кількістю елементів і за точністю в 0,0001.

Завдання №3 VNS Lab 7 - Task 1

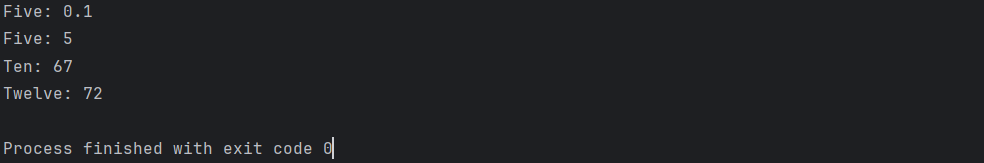


Рисунок : Визначає мінімальне значення серед типу double або int

Завдання №3 VNS Lab 7 - Task 2

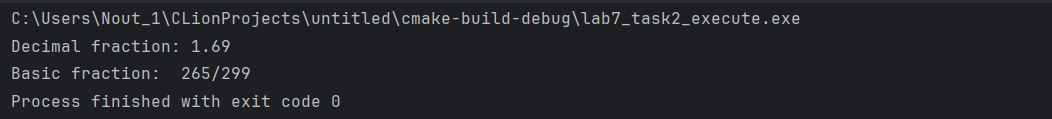


Рисунок : Додає десяткові і прості дроби

Завдання №4 Class Practice Work



Рисунок :Програма здійснює керування бібліотекою

Завдання №5 Self Practice Work.

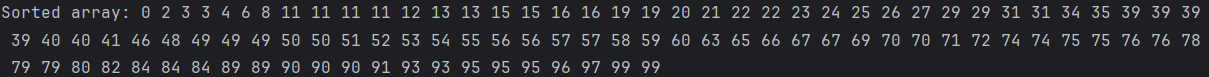
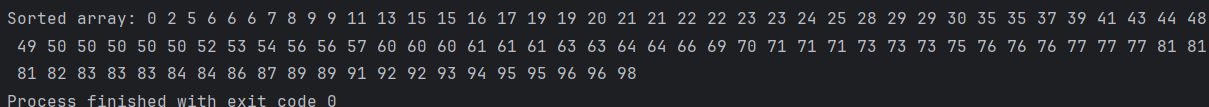


Рисунок 12: Програма сортує масив методом mergesort

# **Висновки:**

За час виконання завдання я навчився працювати з вкладеними циклами, такими як for, while і do while. Я написав завдання з VNS і практичні завдання, в яких використав вкладені цикли. Також я детально познайомився з функціями, навчився їх перевантажувати. Я написав функції для різних завдань, зокрема для обчислення факторіалу числа, для пошуку степенів чисел та для обчислення суми послідовності. Крім того, я попрацював з рекурсією. Я написав рекурсивні функції для обчислення факторіалу числа, для пошуку коренів квадратного рівняння та для обчислення суми послідовності.

Всі зроблені задачі і звіт я запушив на GitHub і зробив pull request.