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:*** «Основи програмування»

до:

ВНС Лабораторної Роботи № 2

ВНС Лабораторної Роботи № 3

ВНС Лабораторної Роботи № 7

Практичних Робіт до блоку № 3

**Виконав:**

Студент групи ШІ-12

Бобровицький Олександр Сергійович

**Тема роботи:** Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції з змінною кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції.

**Мета:** ознайомитись з темами зазначеними вище, опрацювати їх теоретично та навчитися використовувати отриманні знання для вирішення практичних задач

**Теоретичні відомості:**

Тема №1 : Введення в Цикли та їх Види в С++, вкладені цикли:

* Джерела:
  + <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/>
  + <https://acode.com.ua/urok-73-operatory-break-i-continue/>
  + cpp.dp.ua/operatory-tsyklu/
* Що опрацьовано:
  + <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/>
  + <https://acode.com.ua/urok-73-operatory-break-i-continue/>
  + cpp.dp.ua/operatory-tsyklu/
* Статус: Ознайомлений
* Початок опрацювання теми: 01.09
* Звершення опрацювання теми: 21.09

Тема №2 : Основи Функцій у С++, Перевантаження Функцій та Простір Імен,

Розширені Можливості Функцій:

* Джерела:
  + <https://learn.microsoft.com/en-us/cpp/cpp/functions-cpp?view=msvc-170>
  + <https://www.w3schools.com/cpp/cpp_functions.asp>
  + <https://www.geeksforgeeks.org/functions-in-cpp/>
  + https://www.geeksforgeeks.org/function-overloading-c/
* Що опрацьовано:
  + <https://learn.microsoft.com/uk-ua/cpp/cpp/data-type-ranges?view=msvc-160>
  + <https://www.w3schools.com/cpp/cpp_functions.asp>
  + <https://www.geeksforgeeks.org/functions-in-cpp/>
  + https://www.geeksforgeeks.org/function-overloading-c/
* Статус: Ознайомлений
* Початок опрацювання теми: 01.09
* Звершення опрацювання теми: 01.10

**Виконання роботи:**

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

Завдання №1 VNS Lab 2

* 11 варіант
* Деталі завдання:
  + Використовуючи оператор циклу, знайти суму елементів, зазначених у конкретному варіанті. Результат надрукувати, надавши відповідний заголовком.
  + Знайти суму ряду з точністю ε=0.0001, загальний член якого= n!/n^n
* Важливі деталі для врахування в імплементації програми
  + При складанні програми вважати, що точність досягнута, якщо <ε

Завдання №2 VNS Lab 3

* 11 варіант
* Деталі завдання:
  + 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;
    - б) для заданої точності ε (ε=0.0001).
* Для порівняння знайти точне значення функції.
* Важливі деталі для врахування в імплементації програми

Завдання №3 VNS Lab 7 Task 1

* 11 варіант
* Деталі завдання:
  + Написати функцію sum зі змінною кількістю параметрів, що знаходить суму заданих звичайних дробів. Написати викликаючу функцію main, що звертається до функції sum не менше трьох разів з кількістю параметрів 5, 10, 12.
* Важливі деталі для врахування в імплементації програми
  + Розв’язати зазначене у варіанті завдання, використовуючи функції зі змінною кількістю параметрів.

Завдання №4 VNS Lab 7 Task 2

* Деталі завдання
  + Написати перевантажені функції й основну програму, що їх викликає.
    - а) для множення десяткових дробів;
    - б) для множення звичайних дробів.

Завдання №5 **Class Practice Work**

* Варіант завдання 1
* Деталі завдання
* 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.
* Важливі деталі для врахування в імплементації програми
  + 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 each: перевірити наявність кожної книги.
  + goto: якщо користувач вводить неправильний вибір, використовуйте goto, щоб перенаправити його до головного меню.

Завдання №6 **Self Practice Work**

* 0010 варіант
* Деталі завдання:
  + Юний художник Зеник має n послідовно розташованих білих комірок, пронумерованих від 1 до n зліва направо, які можна фарбувати. Він виконує m фарбувань такого вигляду: (l, r, c) — зафарбувати усі комірки від l до r включно в колір c. Кожна комірка пофарбована тільки в останній колір, яким фарбувалася відовідна комірка. Вам потрібно відновити кінцеву картину, тобто визначити кольори усіх комірок після виконання всіх запитів.
  + Input
    - У першому рядку задано два цілих числа n і m— кількість комірок та кількість фарбувань.У наступних m рядках задано по 3 цілих числа l, r і c— відповідне фарбування.
* Output
  + У єдиному рядку виведіть n цілих чисел через пробіл — кольори комірок після усіх фарбувань. Вважайте, що білий колір — це 0.
* Важливі деталі для врахування в імплементації програми
  + Constraints
    - 1 ≤ n, m, c ≤ 10^5
    - 1 ≤ l ≤ r ≤ n .

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

Програма №1 VNS Lab 2

* Блок-схема
* Зображення, що містить текст, знімок екрана, коло, Шрифт

  Автоматично згенерований опис

Планований час на реалізацію – 20min

Програма №2 VNS Lab 3

* Блок-схемаЗображення, що містить текст, знімок екрана, Шрифт, графічний дизайн

  Автоматично згенерований опис

Планований час на реалізацію – 30min

Програма №3 VNS Lab 7 Task 1

* Блок-схемаЗображення, що містить текст, знімок екрана, Шрифт, коло

  Автоматично згенерований опис

Планований час на реалізацію – 25min

Програма №4 VNS Lab 7 Task 2

* Блок-схемаЗображення, що містить текст, знімок екрана, коло, Шрифт

  Автоматично згенерований опис

Планований час на реалізацію – 15min

Програма №5 Class Practice Work

* Блок-схемаЗображення, що містить текст, знімок екрана, графічний дизайн, чорно-білий

  Автоматично згенерований опис

Планований час на реалізацію – 35min

Програма №6 Self-Practice Work

* Блок-схема

Зображення, що містить текст, знімок екрана, Шрифт, графічний дизайн

Автоматично згенерований опис

Планований час на реалізацію – 60min

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

Завдання №1

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/173/commits/5808d20d0c6f9bab51052ecda7bc6c94cb4d4403#diff-78d2576b328c73a1c09cf2f17b0edb6abd758cbddecb20e16813a03b7c4b27ab>

#include<iostream>

#include <cmath>

using namespace std;

int Factorial(int n)

{

int factorial = 1.0;

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

{

factorial \*= i;

}

return factorial;

}

int main()

{

double precision = 0.0001;

long double temp = 0.0;

long double sum = 0.0;

int n = 1;

do

{

temp = Factorial(n)/ pow(n, n);

if (temp < precision)

{

break;

}

sum += temp;

n++;

} while (temp >= precision);

cout << sum;

return 0;

}

Код до завдання №1

Завдання №2

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/173/commits/5808d20d0c6f9bab51052ecda7bc6c94cb4d4403#diff-b04aecc530a61a4707844017a474a8e1a6313bfc3fc1bf3a67627a700d386a23>

#include<iostream>

#include <cmath>

using namespace std;

int Factorial(int n)

{

    int factorial = 1.0;

    for(int i = 1; i <= n; i++)

    {

        if (i == 0)

        {

            factorial = 0;

        }

        factorial \*= i;

    }

    return factorial;

}

double PresetPrecision(float x)

{

    double precision = 0.0001;

    long double temp = 0.0;

    double sum = 0.0;

    int n = 0;

    do

    {

        temp = ((2\*n+1) / Factorial(n)) \* pow(x, 2\*n);

        if (temp < precision)

        {

            break;

        }

        sum += temp;

        n++;

    } while (temp >= precision);

    return sum;

}

double PresetNumber(float x)

{

    long double temp = 0.0;

    double sum = 0.0;

    for(int n = 0; n <= 10; n++)

    {

        temp = ((2\*n+1) / Factorial(n)) \* pow(x, 2\*n);

        sum +=temp;

    }

    return sum;

}

long double ExactValue(float x)

{

    long double FunctionValue = (1 + 2\*x\*x) \* exp(x\*x);

    return FunctionValue;

}

int main()

{

    float a = 0.1;

    float b = 1.0;

    float k = 10.0;

    float step = (b-a)/k;

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

    {

        double UsingPresetPrecion = PresetPrecision(x);

        double UsingPresetNumber = PresetNumber(x);

        long double ExactFunctionValue = ExactValue(x);

        cout << "f(x), x = " << x << "\n - Exact value = " << ExactFunctionValue << "\n - With fixed precision = " << UsingPresetPrecion << "\n - With fixed 'n' = " << UsingPresetNumber << endl;

    }

    return 0;

}

Код до завдання №2

Завдання №3

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/173/commits/5808d20d0c6f9bab51052ecda7bc6c94cb4d4403#diff-32d480e156523004b490c4958c297709d71be11e35bf9529625e5dbcb4166c99>

#include<iostream>

#include<cstdarg>

using namespace std;

void sum(int quantity, ...)

{

    va\_list args;

    va\_start(args, quantity);

    double ResultSum = 0.0;

    int Numerator = 0;

    int Denominator = 0;

    for(int i = 0; i < quantity; i++)

    {

        Numerator = va\_arg(args, int);

        Denominator = va\_arg(args, int);

        ResultSum += static\_cast<double>(Numerator) / Denominator;

    }

    va\_end(args);

    cout << "Сума дробів = " << ResultSum;

}

int main()

{

    sum(5, 1, 2, 1, 3, 1, 4, 1, 5, 1, 6);

    cout << " (При 5 єлементах)" << endl;

    sum(10, 1, 2, 1, 3, 1, 4, 1, 5, 1, 6, 1, 7, 1, 8, 1, 9, 1, 10, 1, 11);

    cout << " (При 10 єлементах)" << endl;

    sum(12, 1, 2, 1, 3, 1, 4, 1, 5, 1, 6, 1, 7, 1, 8, 1, 9, 1, 10, 1, 11, 1, 12, 1, 13);

    cout << " (При 12 єлементах)" << endl;

    return 0;

}

Код до завдання №3

Завдання №4

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/173/commits/5808d20d0c6f9bab51052ecda7bc6c94cb4d4403#diff-95f023e9a69734a6b5b9d98db11c4bebb1ea7415b466c4c578a200159430a02e>

#include<iostream>

#include<cstdarg>

using namespace std;

double Multiplication(int Numerator1, int Denominator1, int Numerator2, int Denominator2)

{

    return static\_cast<double>(Numerator1 \* Numerator2) / (Denominator1 \* Denominator2);

}

double Multiplication(double DecimalFraction1, double DecimalFraction2)

{

    return DecimalFraction1 \* DecimalFraction2;

}

int main()

{

    int Numerator1 = 8;

    int Denominator1 = 33;

    int Numerator2 = 17;

    int Denominator2 = 25;

    double DecimalFraction1 = 3.14;

    double DecimalFraction2 = 11.096;

    cout << "Результат множення звичайних дробів : " << Multiplication(Numerator1, Denominator1, Numerator2, Denominator2) << endl;

    cout << "Результат множення десткових дробів : " << Multiplication(DecimalFraction1, DecimalFraction2) << endl;

    return 0;

}

Код до завдання №4

Завдання №5

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/173/commits/5808d20d0c6f9bab51052ecda7bc6c94cb4d4403#diff-d2a6d842ffd1ddce1c60156576004494468c962f1b2657272206e4e7310f354f>

#include<iostream>

#include<vector>

using namespace std;

int main()

{

    vector<string> BookNames {"1.The lord of the rings", "2.Harry Potter", "3.The little prince", "4.Nineteen eighty four", "5.Don Quixote", "6.Bible"};

    int LibrarySize = BookNames.size();

    vector<string> AvailabilityOfBook(LibrarySize, "Available");

    int Option = 1;

    int SelectedBook = 0;

    int temp = 0;

    cout << "What do you wish to do? \n";

    cout << " 0 - Exit\n";

    cout << " 1 - Borrow a book \n";

    cout << " 2 - Return a book \n";

    cout << " 3 - Get the list of the books\n";

    while(Option != 0)

    {

        do

        {

            cout << "Type in the number of the option that you want to pick: ";

            cin >> Option;

            for(string status : AvailabilityOfBook)

                {

                    if(status == "Available")

                    {

                        temp++;

                    }

                    if(temp == LibrarySize)

                    {

                        cout << "All books are available\n";

                    }

                }

            switch (Option)

            {

            case 1:

            TryAgainBorrow:

                cout << "Select the book for borrowing by typing in its number: \n";

                cin >> SelectedBook;

                if(SelectedBook <= LibrarySize &&  AvailabilityOfBook[SelectedBook - 1] != "Unavailable")

                {

                    cout << "You have borrowed book " << BookNames[SelectedBook - 1] << endl;

                    AvailabilityOfBook[SelectedBook - 1] = "Unavailable";

                }

                else if ( AvailabilityOfBook[SelectedBook - 1] == "Unavailable")

                {

                    cout << "Sorry, this book is already taken.\n";

                }

                else

                {

                    cout << "Invalid option, please try again.\n";

                    goto TryAgainBorrow;

                }

                break;

            case 2:

            TryAgainReturn:

                cout << "Type in the number of the book that you want to return.\n";

                cin >> SelectedBook;

                if(SelectedBook <= LibrarySize &&  AvailabilityOfBook[SelectedBook - 1] == "Unavailable")

                {

                    cout << "You have returned book " << BookNames[SelectedBook - 1] << endl;

                    AvailabilityOfBook[SelectedBook - 1] = "Available";

                }

                else if ( AvailabilityOfBook[SelectedBook - 1] == "Available")

                {

                    cout << "Sorry, this book has been already returned.\n";

                }

                else

                {

                    cout << "Invalid option, please try again.\n";

                    goto TryAgainReturn;

                }

                break;

            case 3:

                cout << "The list of the books: \n";

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

                {

                    cout << BookNames[i] << " -" << AvailabilityOfBook[i] << endl;

                }

                break;

            }

        } while (Option != 0);

    }

    return 0;

}

Код до завдання №5

Завдання №6

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/173/commits/5808d20d0c6f9bab51052ecda7bc6c94cb4d4403#diff-02ec8fa8cee76beab6e9255c5a6baaca8ed54eb56c948310c455a744cb446c3f>

#include <bits/stdc++.h>

using namespace std;

int main()

{

    int m, n, l, r, c = 0;

    cin >> n >> m;

    vector<int> v1(n, 0);

    if (m < 1 || m > 1e5 || n < 1 || n > 1e5)

    {

        return 1;

    }

    for (int i = 0; i < m; i++)

    {

        cin >> l >> r >> c;

        if (l < 1 || r > n || r < l || c < 1 || c > 1e5)

        {

            return 1;

        }

        fill(v1.begin() + l - 1, v1.begin() + r , c);

    }

    for (int x = 0; x < n; x++)

    {

        cout << v1[x] << " ";

    }

    return 0;

}

Код до завдання №6 Варіант програми 1

#include <iostream>

#include <vector>

using namespace std;

int main()

{

    int m, n, l, r, c = 0;

    int count = 0;

    cin >> n >> m;

    if (m < 1 || m > 1e5 || n < 1 || n > 1e5)

    {

        return 1;

    }

    vector<int> cells(n, 0);

    vector<vector<int>> color(m, vector<int>(3));

    for(int i = 0; i < m; i++)

    {

        cin >> l >> r >> c;

        if (l < 1 || r > n || r < l || c < 1 || c > 1e5)

        {

            return 1;

        }

        color[i][1] = l;

        color[i][2] = r;

        color[i][3] = c;

    }

    for(int j = m-1; j >= 0; j--)

    {

        if (count == n)

        {

            break;

        }

        for(int i = color[j][1]; i <= color[j][2]; i++)

        {

            if(cells[i-1] == 0)

            {

                cells[i-1] = color[j][3];

                count++;

            }

        }

    }

    for (int x = 0; x < n; x++)

    {

        cout << cells[x] << " ";

    }

    return 0;

}

Код до завдання №6 Варіант програми 2

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

Завдання №1 Деталі по виконанню і тестуванню програми



Блок №1 Результат виконання завдання №1

Час затрачений на виконання завдання – 20min

Завдання №2 Деталі по виконанню і тестуванню програми

Зображення, що містить текст, знімок екрана, Шрифт, меню

Автоматично згенерований опис Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Блок №1 Результат виконання завдання №2

Час затрачений на виконання завдання – 30min

Завдання №3 Деталі по виконанню і тестуванню програми

Зображення, що містить текст, Шрифт, знімок екрана, типографія

Автоматично згенерований опис

Блок №1 Результат виконання завдання №3

Час затрачений на виконання завдання – 25min

Завдання №4 Деталі по виконанню і тестуванню програми

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

Блок №2 Значення змінних



Блок №2 Результат виконання завдання №4

Час затрачений на виконання завдання – 20min

Завдання №5 Деталі по виконанню і тестуванню програми

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Блок №1 Результат виконання завдання №5

Час затрачений на виконання завдання – 40min

Завдання №6 Деталі по виконанню і тестуванню програми

Зображення, що містить текст, Шрифт, знімок екрана, чорний

Автоматично згенерований опис

Блок №1 Результат виконання завдання №6

Час затрачений на виконання завдання – 60min

**5. Кооперація з командою:**

* Скрін з зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

* Скрін з 2-му коментарями від учасників команди на пул реквесті з Ревю Роботи

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

**Висновок:** я ознайомився з теорією за пунктами зазначеними в темі роботи, опрацював їх та навчився використовувати отриманні знання для вирішення практичних задач.