Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт №3**

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

***з розділу***: «Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія»

***Виконала:***

студентка групи ШІ-13

Кузнєцова Анастасія Володимирівна

# **Тема роботи:**

Засвоєння основних операторів мови С. Обчислення функцій з використанням їхнього розкладу в степеневий ряд. Дослідження функцій зі змінною кількістю параметрів та перевантаження функцій в С++.

# **Мета роботи:**

Отримання навичок у виборі й використанні операторів С. Знайомство з ітераційними процесами. Практика в організації ітераційних й арифметичних циклів. Знайомство з організацією функцій зі змінною кількістю параметрів та перевантаженням функцій.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Параметри і аргументи функцій.
* Тема №2: Цикл while, for, do while.
* Тема №3: Уліпсис.
* Тема №4: Оператор goto.
* Тема №5: Масиви.
* Тема №6: Перевантаження функцій. Вбудовані функції.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Функції, їх параметри та аргументи.
  + Джерела Інформації:
    - Стаття: https://acode.com.ua/urok-102-parametry-i-argumenty-funktsij/  
      <https://acode.com.ua/urok-103-peredacha-po-znachennyu/>
    - Відео: https://youtu.be/G8P6SvdqU9s?si=o0Adu6zYykdMEJpW
  + Що опрацьовано:

Використання параметрів і аргументів функції.

* + Статус: Ознайомлений
  + Початок опрацювання теми: 15.11.23
  + Звершення опрацювання теми: 19.11.23
* Тема №2: Цикл while, for, do while.
  + Джерела Інформації:
    - Відео: https://youtu.be/zBtcqNdiRf4?si=\_qBv-oYewTNHuO9a
    - Стаття: 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/
  + Що опрацьовано:
    - Опрацьовано використання циклів while, do while, for. Їх конструкцію та особливості.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 19.11.23
  + Звершення опрацювання теми: 30.11.23
* Тема №3: Еліпсис.
  + Джерела Інформації:
    - Відео: https://youtu.be/irFkMavpL9A?si=kPwJpBzJaNqiRWHh
    - Стаття: https://acode.com.ua/urok-117-elipsys/
  + Що опрацьовано:

Опрацьовано що таке еліпсис. Опрацьовано використання операторів va\_list, va\_start і va\_end.

* + Статус: Ознайомлений
  + Початок опрацювання теми: 20.11.23
  + Звершення опрацювання теми: 30.11.23
* Тема №4: Оператор goto.
  + Джерела Інформації:
    - Стаття: https://acode.com.ua/urok-69-operator-goto/
  + Що опрацьовано:

Опрацьовано функції оператору goto та як його застосовувати.

* + Статус: Ознайомлений
  + Початок опрацювання теми: 24.11.23
  + Звершення опрацювання теми: 25.11.23
* Тема №5: Масиви. Вектори.
  + Джерела Інформації:
    - Відео: https://youtu.be/ULdbOaMBPYc?si=LDt5a\_HWVW4pB0k\_  
      https://youtu.be/V2g3B9Zbh4Q?si=wkS7PFn21ZEwLK5O
    - Стаття: https://acode.com.ua/urok-77-masyvy/  
      https://acode.com.ua/urok-79-masyvy-i-tsykly/  
      https://acode.com.ua/urok-99-vvedennya-v-std-vector/
  + Що опрацьовано:
    - Опрацьовано як використовувати масиви. Їх конструкцію та особливості.
    - Як прописувати елементи у масиві. Опрацьовано інформацію про вектори та їх використання.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 17.11.23
  + Звершення опрацювання теми: 20.11.23
* Тема №6: Перевантаження функцій. Вбудовані функції.
  + Джерела Інформації:
    - Відео: https://youtu.be/3KJfisev6SI?si=iIo1GyuDy4fCtox1
    - Стаття: https://acode.com.ua/urok-107-vbudovani-funktsiyi/  
      https://acode.com.ua/urok-108-perevantazhennya-funktsij/
  + Що опрацьовано:

Опрацьовано використання перевантаження та вбудованих функцій. Для яких випадків і як його використовувати.

* + Статус: Ознайомлений
  + Початок опрацювання теми: 20.11.23
  + Звершення опрацювання теми: 30.11.23

# **Виконання роботи:**

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

Завдання №1 VNS Lab 2

* Варіант завдання: 8
* Деталі завдання: порахувати і вивести суму ряду за заданим загальним членом.
* Важливі деталі для врахування в імплементації програми: відповідь повинна бути з точністю до епсілон. Варто використовувати рекурентну формулу.

Завдання №2 VNS Lab 3

* Варіант завдання: 8
* 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: для заданого n; для заданої точності ε (ε=0.0001). Для порівняння знайти точне значення функції.
* Важливі деталі для врахування в імплементації програми: Варто звернути увагу на правильний вибір формули для обчислення елемента ряду an і правильне присвоєння початкових значень змінним циклу.

Завдання №3 VNS Lab 7

* Варіант завдання: 8  
  Task 1:
* Деталі завдання:  
  Написати функцію max зі змінною кількістю параметрів, що знаходить максимальне із чисел типу int. Написати викликаючу функцію main, що звертається до функції max.
* Важливі деталі для врахування в імплементації програми: Функція main звертається до функції max не менше трьох разів з кількістю параметрів 5, 10, 12.  
  Task 2:
* Деталі завдання:  
  Написати перевантажені функції й основну програму, що їх викликає. Функція за номером року видає його назву за старояпонським календарем, а за назвою місяця видає знак Зодіаку.
* Важливі деталі для врахування в імплементації програми: У старояпонському календарі кожен з 12-ти послідовних років має назву звіра (пацюк, бик, тигр, заєць, дракон, змія, кінь, вівця, мавпа, півень, собака, кабан), а кожен з 5-ти — має колір (зелений, червоний, жовтий, синій, чорний).

Завдання №4 Class Practice Work

* Деталі завдання: Створити 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.
* Важливі деталі для врахування в імплементації програми: Використати масив або вектор для зберігання назв книг. Використати інший масив або вектор для збереження стану доступності кожної книги.

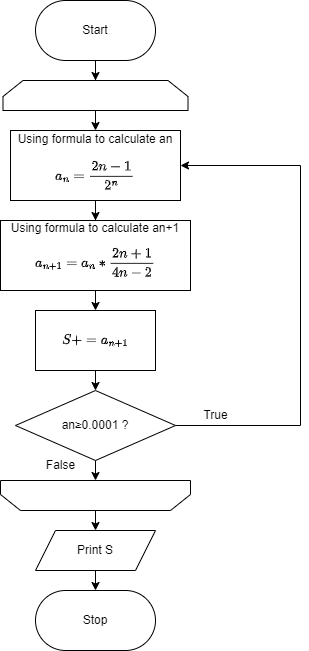
Завдання №5 Self-practice

* Варіант завдання: «Веселі каруселі»
* Деталі завдання: у першому рядку задано два цілих числа k та n — кількість гривень та кількість атракціонів відповідно. У другому рядку задано n цілих чисел cj, j — ціна j-го атракціону. У єдиному рядку вивести одне ціле число — кількість атракціонів, на яких покатається Марічка.
* Важливі деталі для врахування в імплементації програми: звернути увагу на обмеження 100<k, n, cj<1. Спочатку Марічка відвідає четвертий атракціон, потім — другий, а потім перший. Після цього в неї залишиться одна гривня, і цього, на жаль, недостатньо для третього атракціону.

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

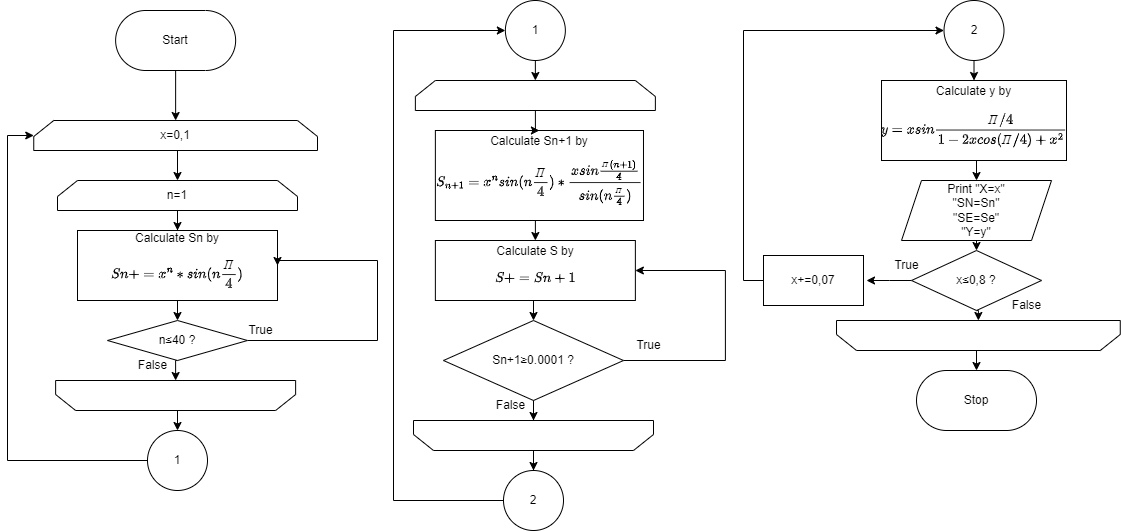
Програма №1 VNS Lab 2

* Планований час на реалізацію: 5 годин
* Важливі деталі для врахування в імплементації: відповідь повинна бути з точністю до епсілон. Варто використовувати рекурентну формулу.

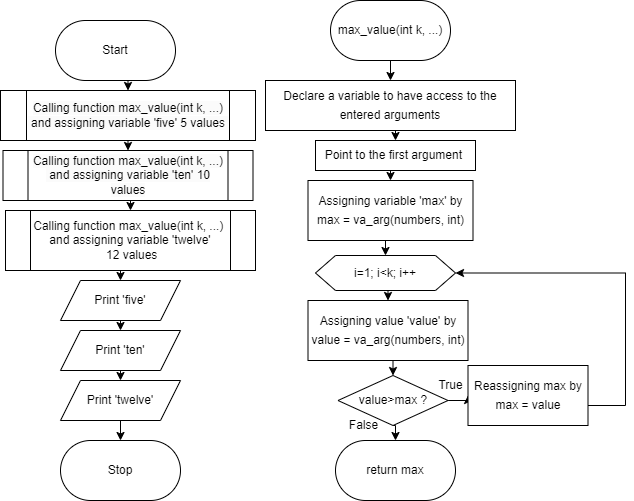


Програма №2 VNS Lab 3

* Планований час на реалізацію: 10 годин
* Важливі деталі для врахування в імплементації: варто звернути увагу на правильний вибір формули для обчислення елемента ряду an і правильне присвоєння початкових значень змінним циклу.

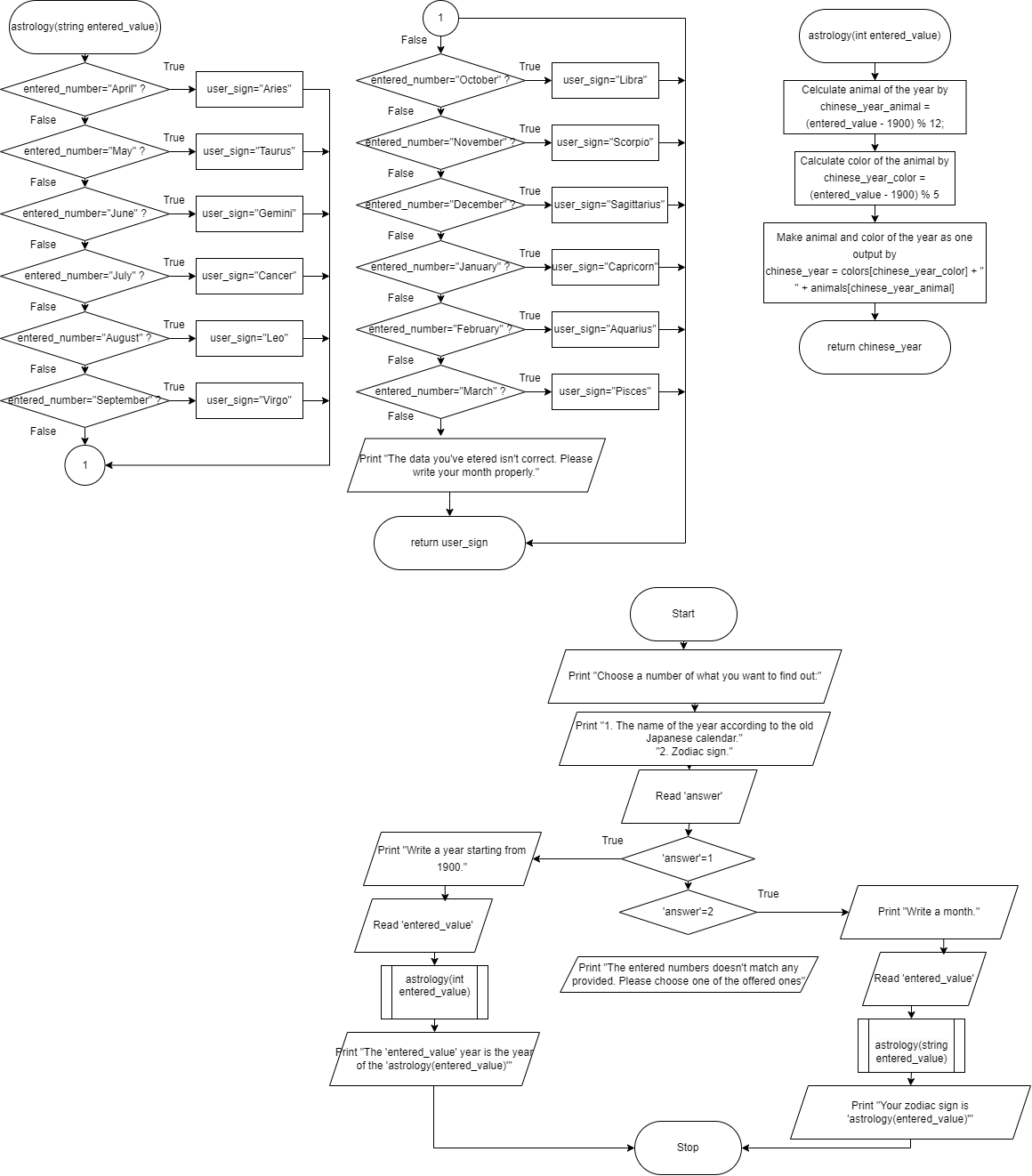


Програма №3 VNS Lab 7 Task 1



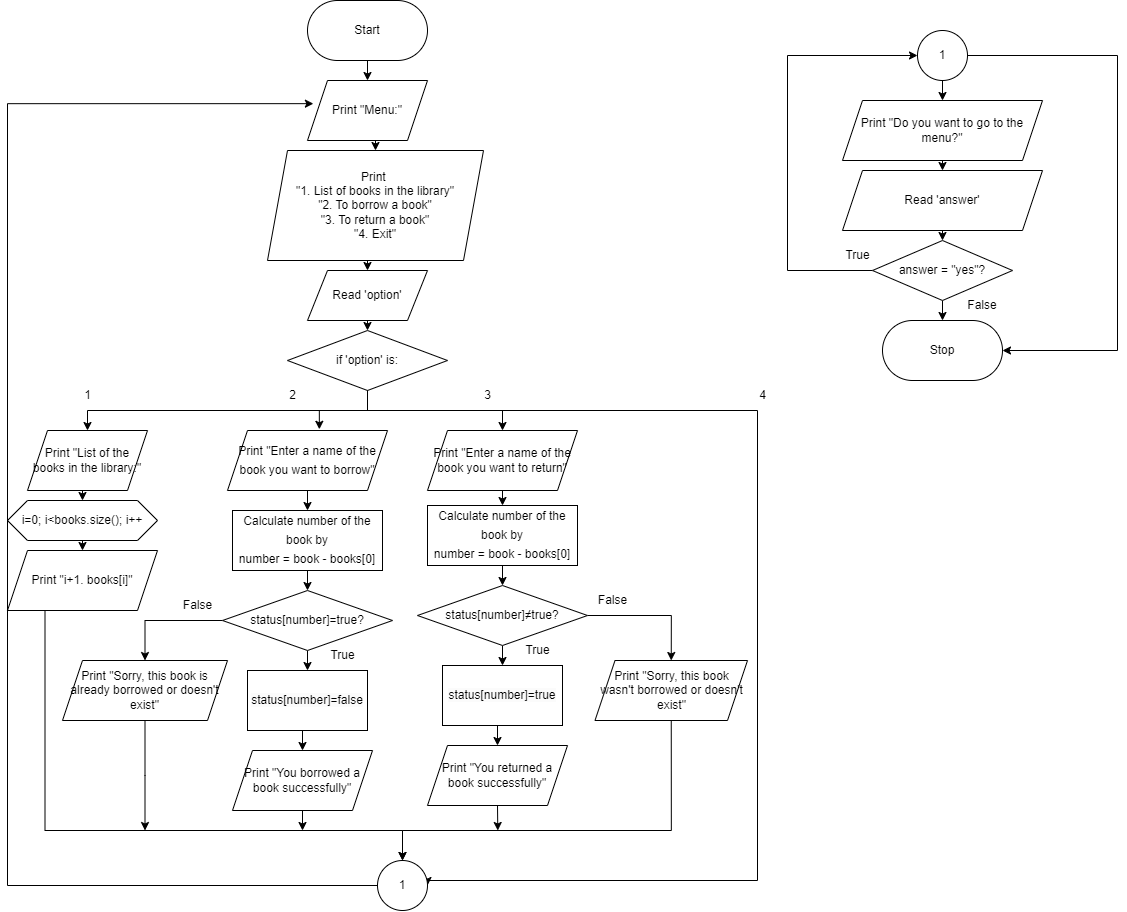
* Планований час на реалізацію: 1 день
* Важливі деталі для врахування в імплементації: Функція main звертається до функції max не менше трьох разів з кількістю параметрів 5, 10, 12.

Програма №4 VNS Lab 7 Task 2



* Планований час на реалізацію: 9 годин
* Важливі деталі для врахування в імплементації: У старояпонському календарі кожен з 12-ти послідовних років має назву звіра (пацюк, бик, тигр, заєць, дракон, змія, кінь, вівця, мавпа, півень, собака, кабан), а кожен з 5-ти — має колір (зелений, червоний, жовтий, синій, чорний).

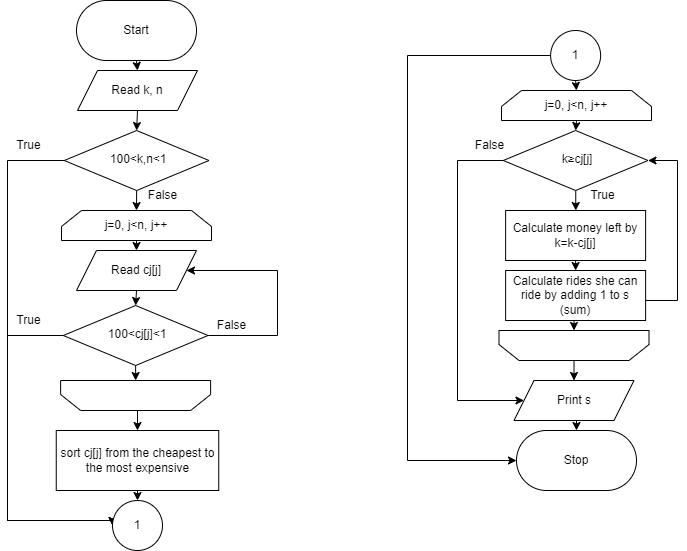
Програма №5 Class Practice Work



* Планований час на реалізацію: 6 годин
* Важливі деталі для врахування в імплементації: Використати масив або вектор для зберігання назв книг. Використати інший масив або вектор для збереження стану доступності кожної книги.

Програма №6 Self-practice

* Планований час на реалізацію: 1 день
* Важливі деталі для врахування в імплементації: звернути увагу на обмеження 100<k, n, cj<1. Спочатку Марічка відвідає четвертий атракціон, потім — другий, а потім перший. Після цього в неї залишиться одна гривня, і цього, на жаль, недостатньо для третього атракціону.



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

Файл settings.json в якому зберігаються налаштування VS Code

{

*"workbench.colorTheme"*: "Monokai",

*"code-runner.runInTerminal"*: true,

*"C\_Cpp.default.compilerPath"*: "d:\\C C++\\test.c",

*"cmake.configureOnOpen"*: true,

*"terminal.integrated.defaultProfile.windows"*: "Windows PowerShell",

*"terminal.integrated.profiles.windows"*: {

*"PowerShell"*: {

*"source"*: "PowerShell",

*"icon"*: "terminal-powershell"

        },

*"Command Prompt"*: {

*"path"*: [

                "${env:windir}\\Sysnative\\cmd.exe",

                "${env:windir}\\System32\\cmd.exe"

            ],

*"args"*: [],

*"icon"*: "terminal-cmd"

        },

*"Git Bash"*: {

*"source"*: "Git Bash"

        },

*"bash (MSYS2)"*: {

*"path"*: "C:\\msys64\\usr\\bin\\bash.exe",

*"args"*: [

                "--login",

                "-i"

            ]

        },

*"Windows PowerShell"*: {

*"path"*: "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe"

        }

    },

*"workbench.colorCustomizations"*: {}

}

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

Завдання №1 VNS Lab 2

#include <iostream>

#include <cmath>

using *namespace* std;

*int* main()

{

*int* n=1;

*double* an=(2\*n-1)/pow(2,n), S;

    do

    {

    an \*= (2\*n+1)/(4\*n-2);

    S+=an;

    n++;

    }

    while  (an>=0.0001);

    cout<<S;

return 0;

}

Завдання полягало у визначенні та виведенні суми ряду на основі заданого загального члена у варіанті.

Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/555/files#diff-d91054761ae315ca84075a4dbcf7124f4627e8488b1f3d6943dc4c27bdcd7896

Завдання №2 VNS Lab 3

#include <iostream>

#include <cmath>

using *namespace* std;

*int* main ()

{

for(*double* x=0.1; x<=0.8; x+=0.07)

{

*double* Sn=0;

    for (*int* j=1; j<=40; j++)

    {

        Sn+=pow(x,j)\*sin(j\*M\_PI/4);

    }

*double* S=0, Se=0;

*int* n=1;

    S=pow(x,n)\*sin(n\*M\_PI/4);

    while (S>=0.0001)

    {

        S\*=(x\*sin((M\_PI\*(n+1))/4))/sin(n\*M\_PI/4);

        Se+=S;

        n++;

    }

*double* y = x\*sin(M\_PI/4)/(1-2\*x\*cos(M\_PI/4)+pow(x,2));

    cout<<"X="<<x<<" SN="<<Sn<<" SE="<<Se<<" Y="<<y<<"\n";

}

    return 0;

}

Завдання полягає у заміні функції степеневим рядом та обчисленні суми, що включає різні параметри для підсумовування змінної х. Кожен член суми залежить від параметра х та номера n, що визначає позицію цього члена у сумі. Треба було використати змінну x, яка змінюється в діапазоні від a до b з кроком (b-a)/k, де k=10. Код обчислює значення функції f(x) у двох випадках: за заданим значенням n та за заданою точністю ε (ε=0.0001). Також треба знайти точне значення функції для порівняння результатів.

Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/555/files#diff-5cbb6b63fbb349b373c79b269a8191433ee5707733dd32927852af67105021fc

Завдання №3 VNS Lab 7

Task 1

#include <iostream>

#include <cstdarg>

using *namespace* std;

*int* max\_value(*int* *k*, ...)

{

    va\_list numbers;

    va\_start(numbers, *k*);

*int* max = va\_arg(numbers, *int*);

    for (*int* i=1; i<*k*; ++i) {

*int* value = va\_arg(numbers, *int*);

        if (value>max) {

            max = value;

        }

    }

    va\_end(numbers);

    return max;

}

*int* main()

{

*int* five = max\_value(5, 3, 8, 15, 6, 2);

*int* ten = max\_value(10, 19, 8, 3, 17, 0, 2, 55, 31, 50, 30);

*int* twelve = max\_value(12, 44, 61, 21, 1, 5, 8, 3, 10, 92, 52, 17, 87);

    cout<<five<<"\n";

    cout<<ten<<"\n";

    cout<<twelve<<"\n";

    return 0;

}

Програма використовує va\_list, va\_start, va\_arg, і va\_end для знаходження максимального, серед введених, значення у max\_value. Функція отримує кількість аргументів k та порівнює чи значення number в циклі більше ніж зазначене max. Якщо так, то воно записує це значення у max і продовжує цикл. Якщо ні, то цикл завершується і функція повертає найбільше значення. У main вводиться 5, 10 та 12 значень. Для обчислення найбільшого з кожного використовується функція max\_value. Намагаючись використати вказівники, програма поводилась невизначено. Відповідь виводилась різна після кожного програвання програми, тому я використала варіативні функції.

Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/555/files#diff-d08cabc8505c6869956e47b352a271ac9dff65ed955646beebac3db5bb9ac74f

Task 2

#include <iostream>

#include <string>

#include <cmath>

using *namespace* std;

string astrology(*int* *entered\_value*)

{

    const *char*\* animals[]={"Rat","Ox","Tiger","Rabbit","Dragon","Snake","Horse","Sheep","Monkey","Rooster","Dog","Boar"};

    const *char*\* colors[]={"Green","Red","Yellow","Blue","Black"};

*int* chinese\_year\_animal = (*entered\_value* - 1900) % 12;

*int* chinese\_year\_color = (*entered\_value* - 1900) % 5;

    string chinese\_year = string(colors[chinese\_year\_color]) + " " + string(animals[chinese\_year\_animal]);

return chinese\_year;

}

string astrology(string *entered\_value*)

{

    const *char*\* user\_sign;

    const *char*\* sign[]={"Aries","Taurus","Gemini","Cancer","Leo","Virgo","Libra","Scorpio","Sagittarius","Capricorn","Aquarius","Pisces"};

    if (*entered\_value*=="April")

    {

        user\_sign=sign[0];

    }

    else if (*entered\_value*=="May")

    {

        user\_sign=sign[1];

    }

    else if (*entered\_value*=="June")

    {

        user\_sign=sign[2];

    }

    else if (*entered\_value*=="July")

    {

        user\_sign=sign[3];

    }

    else if (*entered\_value*=="August")

    {

        user\_sign=sign[4];

    }

    else if (*entered\_value*=="September")

    {

        user\_sign=sign[5];

    }

    else if (*entered\_value*=="October")

    {

        user\_sign=sign[6];

    }

    else if (*entered\_value*=="November")

    {

        user\_sign=sign[7];

    }

    else if (*entered\_value*=="December")

    {

        user\_sign=sign[8];

    }

    else if (*entered\_value*=="January")

    {

        user\_sign=sign[9];

    }

    else if (*entered\_value*=="February")

    {

        user\_sign=sign[10];

    }

    else if (*entered\_value*=="March")

    {

        user\_sign=sign[11];

    }

    else

    {

        cout<<"The data you've etered isn't correct. Please write your month properly.";

        return 0;

    }

return user\_sign;

}

*int* main()

{

*int* answer;

    cout<<"Choose a number of what you want to find out:\n";

    cout<<"1. The name of the year according to the old Japanese calendar."<<"\n2. Zodiac sign.\n";

    cin>>answer;

    if(answer==1)

    {

*int* entered\_value;

        cout<<"Write a year starting from 1900.\n";

        cin>>entered\_value;

        cout<<"The "<<entered\_value<<" year is the year of the "<<astrology(entered\_value)<<"\n";

    }

    else if(answer==2)

    {

        string entered\_value;

        cout<<"Write a month.\n";

        cin>>entered\_value;

        cout<<"Your zodiac sign is "<<astrology(entered\_value)<<"\n";;

    }

    else

    {

        cout<<"The entered numbers doesn't match any provided. Please choose one of the offered ones";

        return 0;

    }

    return 0;

}

Завдання полягає у тому, щоб використовуючи функції змінною кількістю параметрів, написати перевантажені функції та основну програму, яка їх викликає. У першому випадку, функція приймає номер року і повертає його назву за старояпонським календарем. У другому випадку, функція отримує назву місяця і повертає відповідний знак Зодіаку.  
Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/555/files#diff-0e2e8f5855d197ede78c9ad161a663b823d13b606f6b2afc865300577b9b695b

Завдання №4 Class Practice Work

#include <iostream>

#include <cstdio>

#include <vector>

#include <string>

using *namespace* std;

*int* main ()

{

    vector<string> books = {"The Seven Husbands of Evelyn Hugo","I see you are interested in the dark","The Maid","A silent witness","Harry Potter"};

    string name, answer;

    vector <*bool*> status = {false, true, false ,true, true};

*int* option;

menu:

    cout<<"\nMenu:\n";

    cout<<"1. List of books in the library\n";

    cout<<"2. To borrow a book\n";

    cout<<"3. To return a book\n";

    cout<<"4. Exit\n";

    cin>>option;

switch(option)

    {

    case 1:

        cout<<"List of the books in the library:\n";

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

        {

            cout<<i+1<<". "<<books[i]<<"\n";

        }

    break;

    case 2:

    cin.ignore();

        cout<<"Enter a name of the book you want to borrow\n";

        getline(cin,name);

        for(const *auto*&book : books)

        {

            if (book == name)

            {

*int* number = &book - &books[0];

                if (status[number])

                {

                    status[number]=false;

                    cout<<"You borrowed a book successfully";

                }

                else

                {cout<<"Sorry, this book is already borrowed or doesn't exist";}

            break;

            }

        }

    break;

    case 3:

    cin.ignore();

        cout<<"Enter a name of the book you want to return\n";

        getline(cin,name);

        for(const *auto*&book : books)

        {

            if (book == name)

            {

*int* number = &book - &books[0];

                if (!status[number])

                {

                    status[number]=true;

                    cout<<"You returned a book successfully";

                }

                else{cout<<"Sorry, this book wasn't borrowed or doesn't exist";}

            break;

            }

        }

    break;

    case 4:

        return 0;

    }

cout<<"\nDo you want to go to the menu?\n";

cin>>answer;

if (answer == "yes" || answer == "Yes" || answer == "YES")

{

    goto menu;

}

else

{

    return 0;

}

return 0;

}

Завданням було створити програму керування бібліотекою, де книги можна було позичати або повертати. Основні функціональності програми включали перелік всіх книг, можливість взяти книгу (за умови наявності) та повернути книгу. У виконанні завдання використовувалися такі структури даних, як масив або вектор для зберігання назв книг, а також інший масив або вектор для відстеження стану доступності кожної книги. Для реалізації циклу в програмі використовувалися конструкції while, do-while та for. Використання цих циклів забезпечувало продовження виконання програми до того моменту, поки користувач не вирішить вийти. Після кожної операції (позичити, повернути, перерахувати), за допомогою do-while, запитувалося у користувача, чи він бажає виконати ще яку-небудь операцію. Для виведення списку усіх книг застосовувався цикл for, а для перевірки наявності кожної книги - цикл for each. Якщо користувач хоче повернутись до меню, то код пересилає його назад до початкового меню за допомогою оператора togo.

Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/555/files#diff-a9d3f9c154ba172e49794f547d8dcc2fff2887dfbdb661af2f5bf7f2d77d17b7

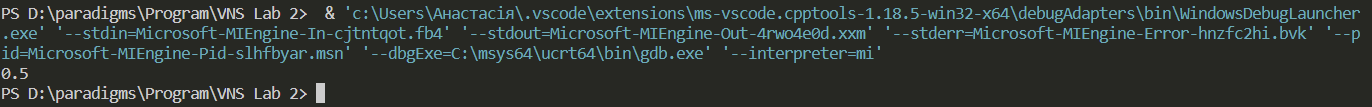
Завдання №5 Self-practice

* #include <iostream>
* #include <vector>
* #include <algorithm>
* using *namespace* std;
* *int* main()
* {
* *int* k, n, s;
* //money = k ; ride = n ; price = cj ; s = rides Marichka can ride on
* cin>>k>>n; //number of hryvnias and number of amusement rides
* if (k<1||n<1||k>100||n>100)
* {
* return 0;
* }
* vector<*int*> cj(n);
* for (*int* j=0; j<n; j++)
* {
* cin>>cj[j]; //the price of the amusement ride
* if (1>cj[j] || cj[j]>100)
* {
* return 0;
* }
* }
* sort(cj.begin(), cj.end());
* for(*int* j=0; j<n; j++)
* {
* if (k>=cj[j])
* {
* k-=cj[j];
* s++;
* }
* else
* {
* break;
* }
* }
* cout<<s; //the total number of rides Marichka can ride on
* return 0;
* }

Завдання полягає у тому, щоб вивести одне ціле число — кількість атракціонів, на яких покатається Марічка. У першому рядку вводиться кількість грошей Марічки та загальна кількість атракціонів у парку. Якщо введені данні не входять у обмеження програма завершує роботу. Далі вводжу вектор, для подальшого сортування цін атракціонів від найдешевшого до найдорожчого. Ціни на атракціони вводять у масив і, якщо вони не входять у обмеження, програма завершує роботу. Використовую цикл і, якщо кількість грошей у Марічки не менше за ціну атракціону, додається 1 атракціон до суми, якщо ні, то цикл завершується. Виводиться кінцева кількість атракціонів, які дівчинка може відвідати.  
Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/555/files#diff-741c9b93aaec8bac43eb2deea57c92ed0d4bd3b73f998205bac5d8d5649c47c9

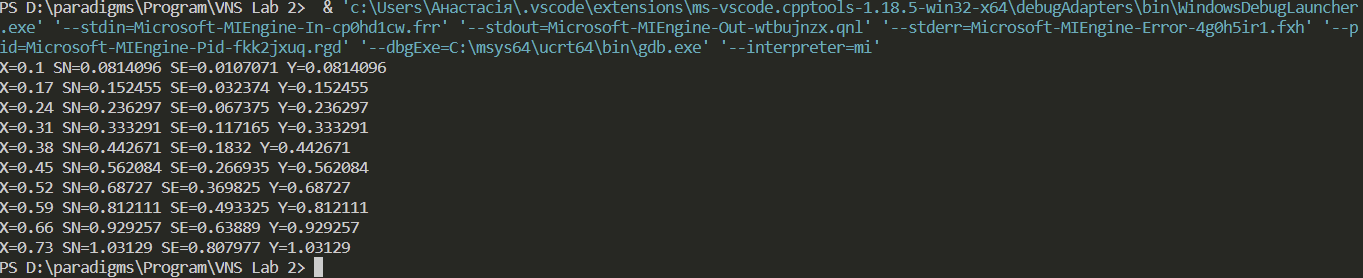
## **5. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 VNS Lab 2



Програма виконала обчислення та вивела суму ряду, використовуючи заданий загальний член. За допомогою оператора циклу програма знайшла суму елементів ряду з точністю ε=0.0001. Для визначення суми членів ряду використовувалася рекурентна формула для отримання наступного члена ряду. Точність вважається досягненою, якщо значення аn стає меншим за ε.  
Час, затрачений на виконання завдання: 2 дні

Завдання №2 VNS Lab 3

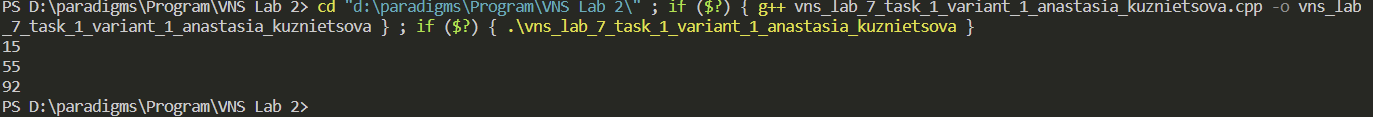


Програма виконала завдання заміни функції степеневим рядом та обчислення суми, включаючи різні параметри для підсумовування змінної x. Кожен член суми залежить від параметра x та номера n, що визначає позицію цього члена у сумі. Цикл відбувається до того момента поки x не стане менше рівне 0,8. X змінюється на значення 0,07 вирахуване за формулою (b-a)/k (k=10). Код обчислює значення функції f(x) у двох випадках: за заданим значенням n та за заданою точністю ε (ε=0.0001). Останнє виведене значення це значення функції для подальшого порівняння результатів.

Час, затрачений на виконання завдання: 3 дні

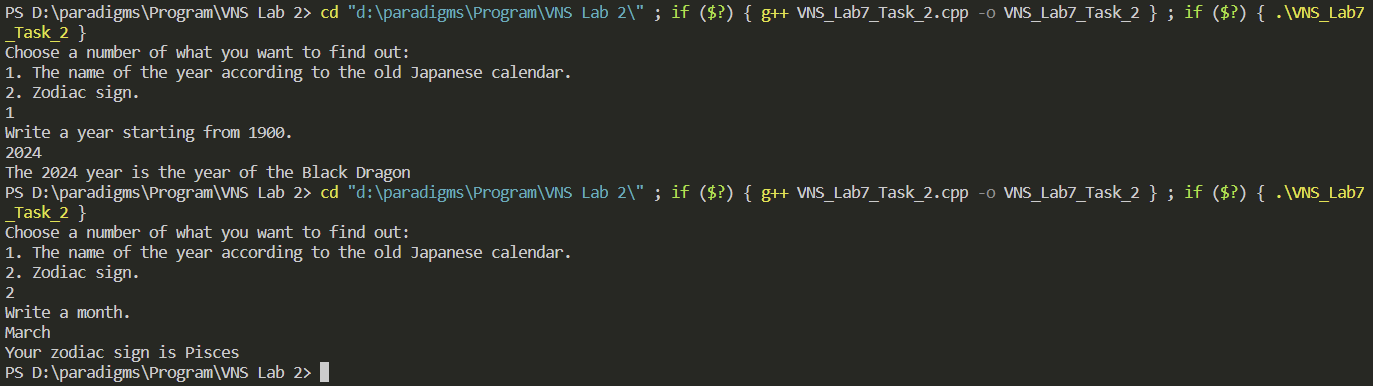
Завдання №3 VNS Lab 7

Task 1



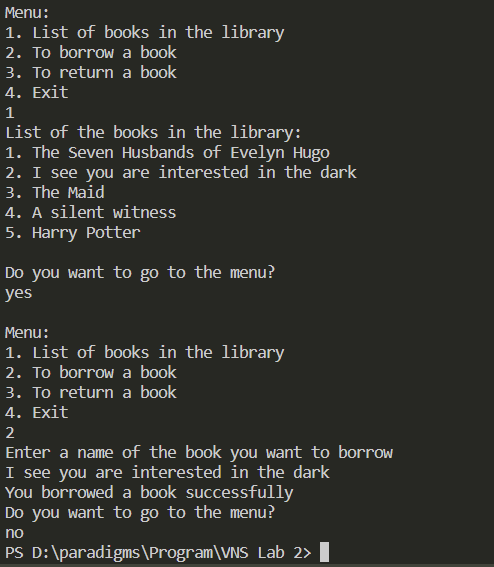
Програма виконала обчислення максимального значення серед введених 5, 10 та 12 значень у функції main. Після порівняння чисел 5, 3, 8, 15, 6, 2 програма видала 15 як найбільше. Відповідно з числами 10, 19, 8, 3, 17, 0, 2, 55, 31, 50, 30 та 12, 44, 61, 21, 1, 5, 8, 3, 10, 92, 52, 17, 87.

Час, затрачений на виконання завдання: 3 дні

Task 2

Створена програма використовує перевантажені функції та основну програму, яка їх викликає. Функції працюють так, що за номером року вони повертають його назву відповідно до старояпонського календаря, а за назвою місяця – визначають відповідний знак Зодіаку. Так як користувач перше вводить значення int, то компілятор перекидає до першої функції і введеному числу 2024, програма виводить "Чорний Дракон". Коли користувач вводить значення string компілятор видає функцію, за допомогою якої програма виводить знак зодіаку в залежності від введеного місяця.  
Час, затрачений на виконання завдання: 2 дні

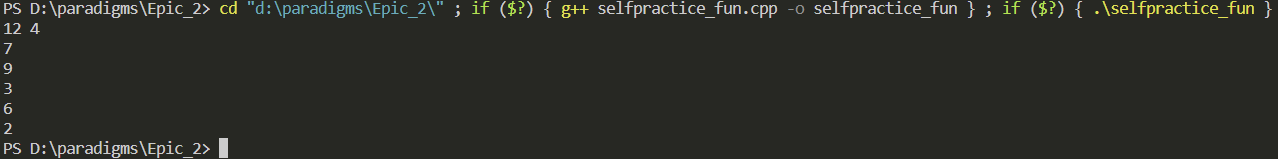
Завдання №4 Class Practice Work



Програма дозволяє керувати бібліотекою: викликати список доступної літератури, повертати та брати книжки. Спочатку вона відобразила список доступних книг, а потім повернула користувача до головного меню, оскільки він ввів "yes" після питання «Чи хочете ви повернутися до меню». Далі користувач обрав позичити книгу. Користувач ввів назви книги, яку хоче позичити, і компілятор перевірив, чи ця книга в даний момент доступна в бібліотеці. Так як вона була доступна, то користувач зміг позичити цю книгу. Далі програма змінила статус цієї книги на "недоступна".

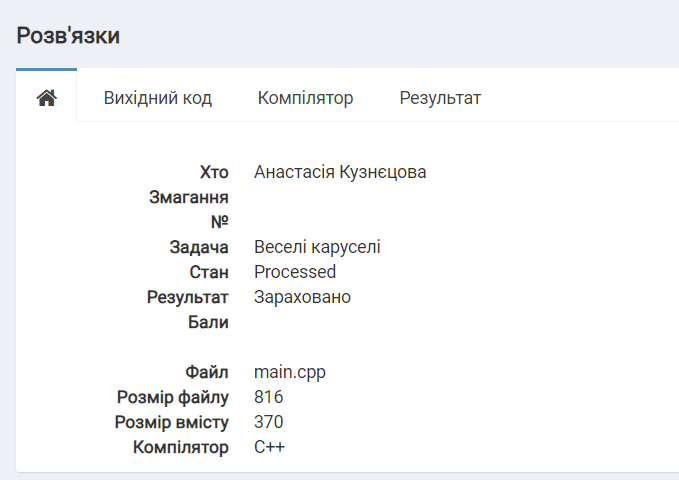
Час, затрачений на виконання завдання: 1 день

Завдання №5 Self-practice



Вхідними даними є: 12 – гроші, які є у Марічки, 4 – кількість атракціонів, які є в парку. Далі 7, 9, 3, 6 – ціни на кожну гірку. Відповідь 2 атракціони, бо Марічка спочатку сходить на найдешевшу і далі по зростанню ціни. Так, як вона спочатку піде на гірку за 3, потім 6 вона вже витратить 9 гривень, а для наступної гірки треба 7.

Скріншот з алготестера:



Час, затрачений на виконання завдання: 1 день

# **Pull-request:** https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/555

# **Висновки:**

Отже, були опрацьовані оператори мови C та їх вивчене використання. Також освоєно ітераційні й арифметичні цикли, та використано їх на практиці. Опрацьовано функцію зі змінною кількістю параметрів та перенавантаження функцій.