

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: «Основи програмування»

до:

ВНС Лабораторної Роботи № 2
ВНС Лабораторної Роботи № 3
ВНС Лабораторної Роботи № 7
Практичних Робіт до блоку № 3

Виконав:

Студент групи ШІ-11
Станько Олег Ігорович

Тема роботи:

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

Мета роботи:

Ознайомитися з основами циклів, дослідити концепцію функцій у C++, навчитися зображати функції в блок схемах, розібратися зі способами завершення виконання циклів.

Теоретичні відомості:

Тема №1: Введення в Цикли та їх Види в C++:

Тема №2: Управління Виконанням Циклів:

Тема №3: Вкладені Цикли:

Тема №4: Основи Функцій у C++:

Тема №5: Перевантаження Функцій та Простір Імен:

Тема №6: Розширені Можливості Функцій:

Тема №7: Вбудовані Функції в C++:

Індивідуальний план опрацювання теорії:

Тема №1: Введення в Цикли та їх Види в C++:

Вивчив такі види циклів: for, while, do-while.

[C++. Цикли. Оператори циклу for, while, do...while | BestProg](#)

Витратив 10 хвилин

Тема №2: Управління Виконанням Циклів:

Ознайомився з exit(), goto, break.

[Оператори управління потоком виконання програм в C++ / aCode](#)

Витратив 5 хвилин

Тема №3: Вкладені Цикли:

Ознайомився з вкладеними циклами.

[C++ • Теорія • Урок 38 • Вкладені конструкції](#)

Витратив 7 хвилин

Тема №4: Основи Функцій у C++:

Ознайомився з функціями, зрозумів як їх зображати в блок схемах

[Функції в C++ / aCode](#)

Витратив 20 хвилин

Тема №5: Перевантаження Функцій та Простір Імен:

Ознайомився з перевантаженням функцій.

[Перевантаження функцій в C++ / Уроки по C++ / aCode](#)

Витратив 7 хвилин

Тема №6: Розширені Можливості Функцій:

Ознайомився з функцією зі змінною кількістю параметрів, областю видимості функції – static, extern, рекурсією, передачею масивів та об'єктів як параметрів, поверненням масивів та об'єктів з функцій.

[Еліпсис в C++ / Уроки по C++ / aCode](#)

Витратив 2 години

Тема №7: Вбудовані Функції в C++:

Ознайомився з вбудованими функціями

[Вбудовані функції в C++ / Уроки по C++ / aCode](#)

Витратив 1 годину

Виконання роботи:

1. Опрацювання завдання та вимог до програм.

VNS LAB 2

Використовуючи оператор циклу, знайти суму елементів, зазначених у конкретному варіанті. Результат надрукувати, надавши відповідний заголовок.

Знайти суму ряду з точністю $\epsilon=0.0001$, загальний член якого $a_n = \frac{n^3}{(3n-3)!}$

VNS LAB 3

Для x , 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 ($\epsilon=0.0001$). Для порівняння знайти точне значення функції

функція	Діапазон зміни аргументу	n	Сума
$y = \frac{1}{2} - \frac{\pi}{4} \sin x $	$0.1 \leq x \leq 0.8$	50	$S = \frac{\cos 2x}{3} + \frac{\cos 4x}{15} + \dots + \frac{\cos 2nx}{4n^2 - 1}$

VNS LAB 7 TASK 1

Написати функцію `пок` зі змінною кількістю параметрів, що знаходить найменше спільне кратне для декількох чисел. $\text{Нск}(a, b) = \frac{a*b}{\text{НСД}(a,b)}$ (НСД- найбільший спільний дільник)

Написати викликаючу функцію `main`, що звертається до функції `пок` не менше трьох разів з кількістю параметрів 3, 5, 6.

VNS LAB 7 TASK 2

Написати перевантажені функції й основну програму, що їх викликає.

а) для масиву цілих чисел знаходить максимальний елемент;

б) для рядка знаходить довжину найдовшого слова .

practice work

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.

Мета Задачі

Навчитися користуватися операторами циклів та функцією переходу на мітку:

1. `for() { ... }`
2. `for each`
3. `while() { ... }`
4. `do { ... } while()`
5. `go to`

Вимоги:

1. `while`: продовжувати працювати, доки користувач не вирішить вийти.
2. `do while`: Після кожної операції (позичити, повернути, перерахувати) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.
3. `for`: список усіх книг за допомогою циклу.
4. `for each`: перевірити наявність кожної книги.
5. `goto`: якщо користувач вводить неправильний вибір, використовуйте `goto`, щоб перенаправити його до головного меню.

Self-practice task

Останнім часом Зеник полюбив викладати портрети визначних історичних постатей України, а також крилаті фрази із ... незвичних матеріалів. Портрет можна уявити як прямокутну сітку розміру n на m , де клітинки позначені «.», якщо вони повинні бути порожніми і «*», якщо у них повинен лежати ... незвичний матеріал. Оскільки один шматок незвичного матеріалу досить важкий (в середньому кілограмів 74), Зеник підвозить матеріали вантажівкою.

За один заїзд вантажівки Зеник може підвезти від 1 до k шматків матеріалу. Вантажівка досить неповоротка, тому за один заїзд здатна підвозити матеріали лише до одного рядка. Тобто перед тим, як вивантажити чергові матеріали, Зеник вибирає один рядок сітки і вивантажує всі матеріали саме в тому рядку. Тоді Зеник знову їде за матеріалами, знову вибирає рядок і знову вивантажує всі матеріали у вибраному рядку і так доти не викладе, те що він початково запланував.

Скільки поїздок вантажівкою необхідно Зенику, щоб викласти портрет?

Вхідні дані

У першому рядку задано три числа n , m і k — розміри сітки і кількість незвичних матеріалів, які Зеник здатен перевести за один раз у один рядок.

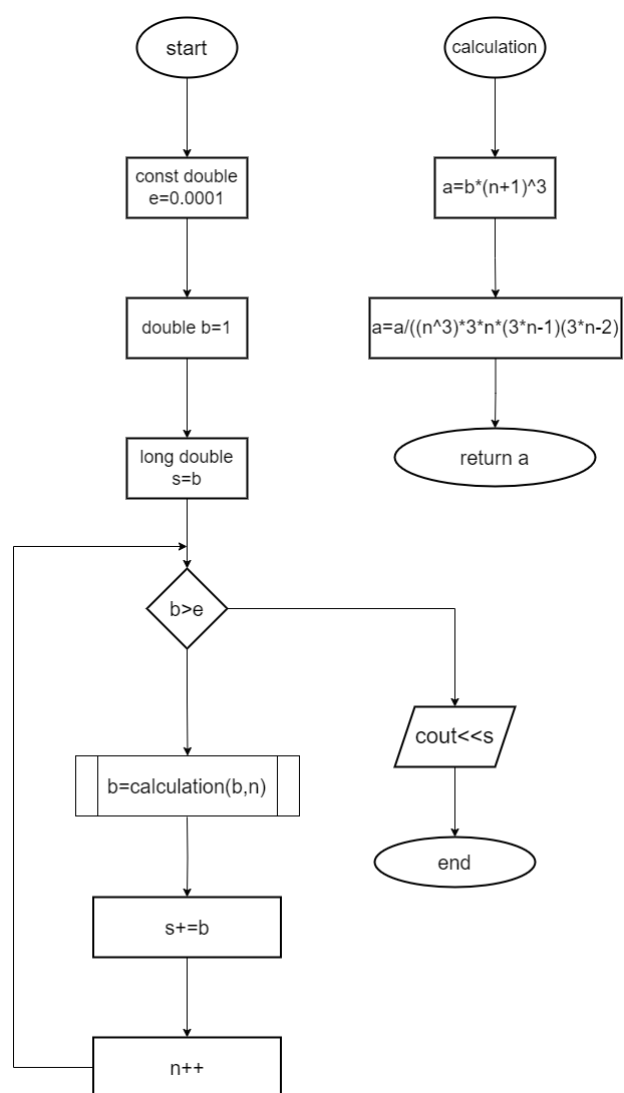
У кожному з наступних n рядків задано сітку, яка складається із символів «.» і «*».

Вихідні дані

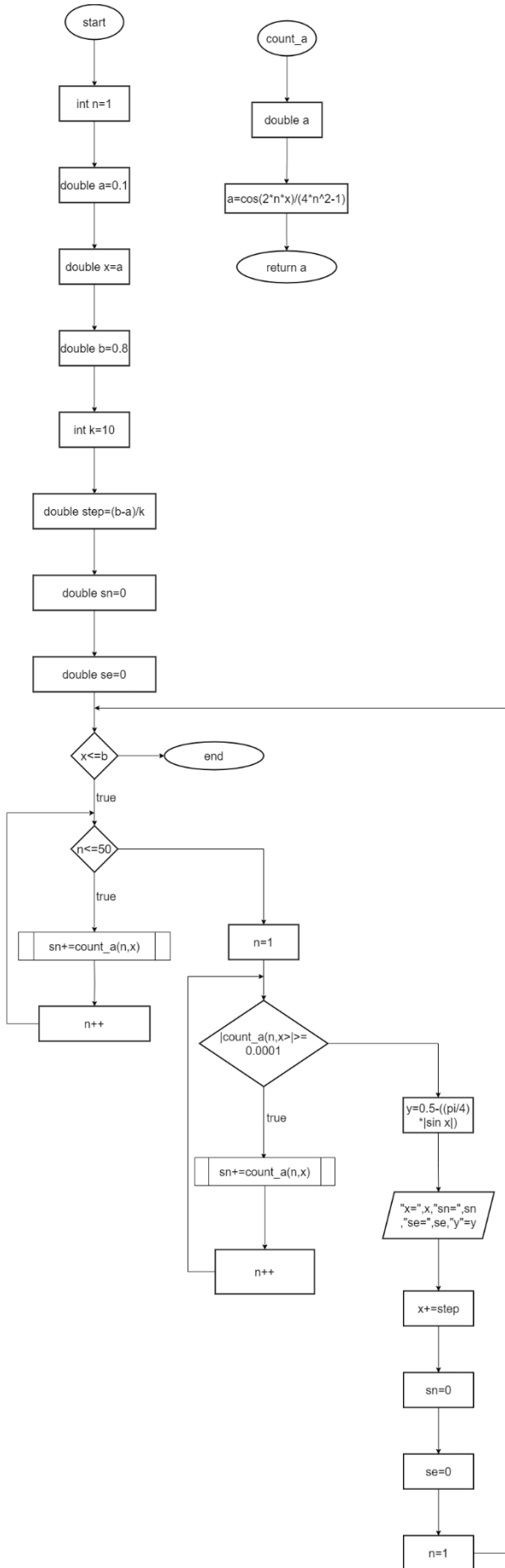
Виведіть одне число — мінімальну кількість поїздок вантажівки, яку повинен здійснити Зеник, для того, щоб викласти портрет.

2. Блок-схеми

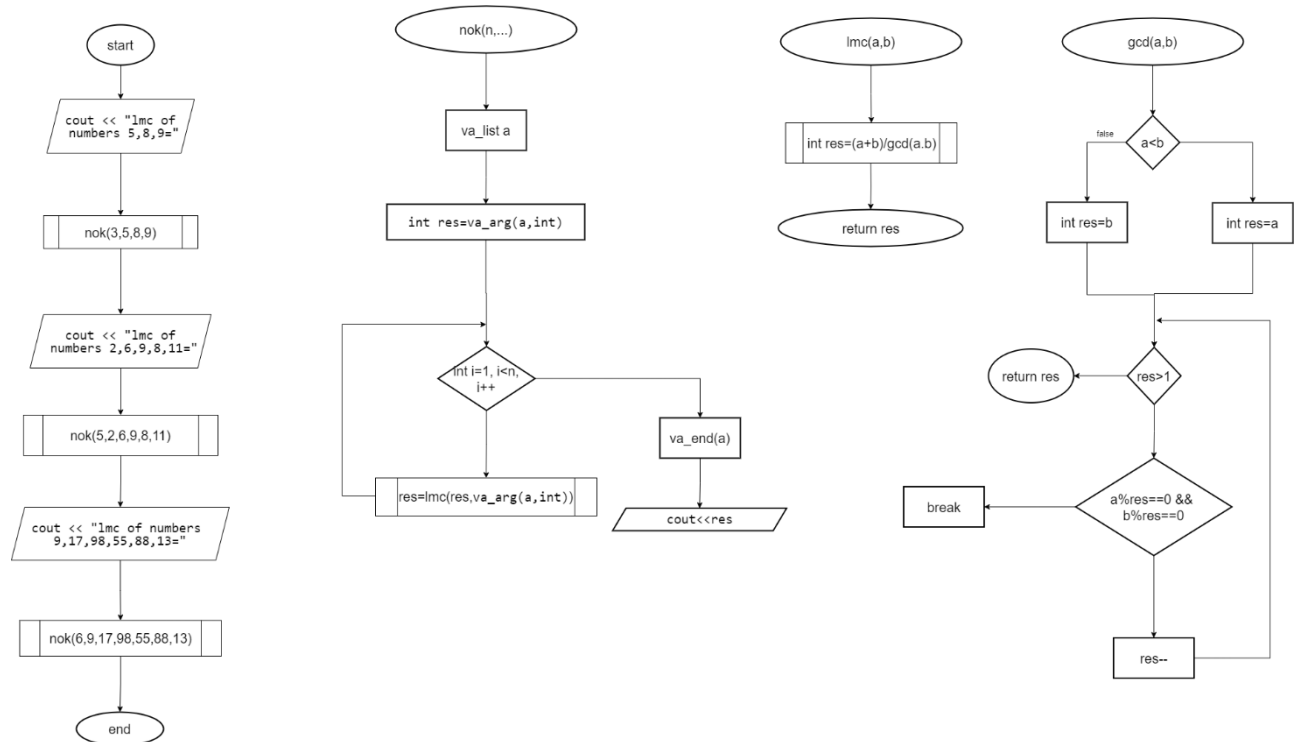
VNS LAB 2



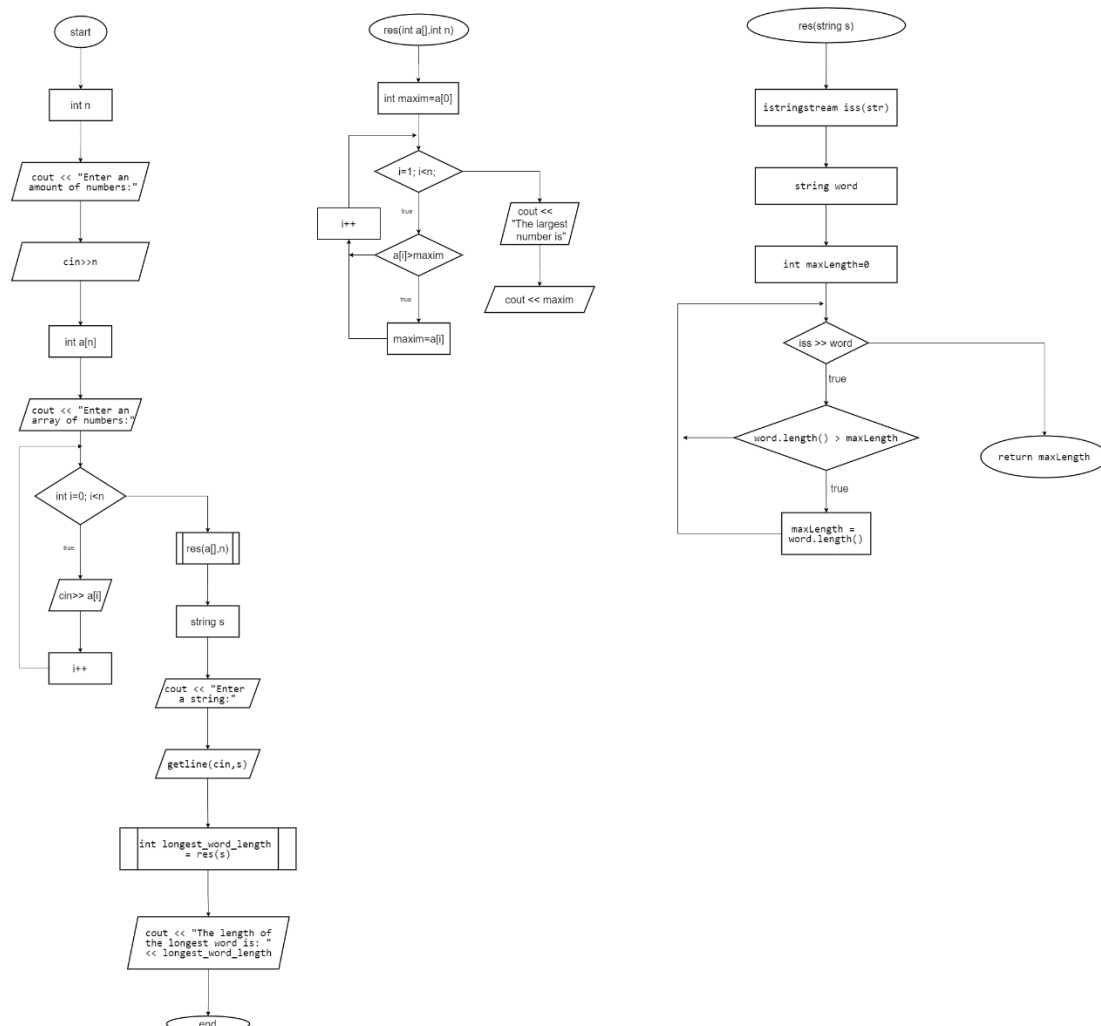
VNS LAB 3



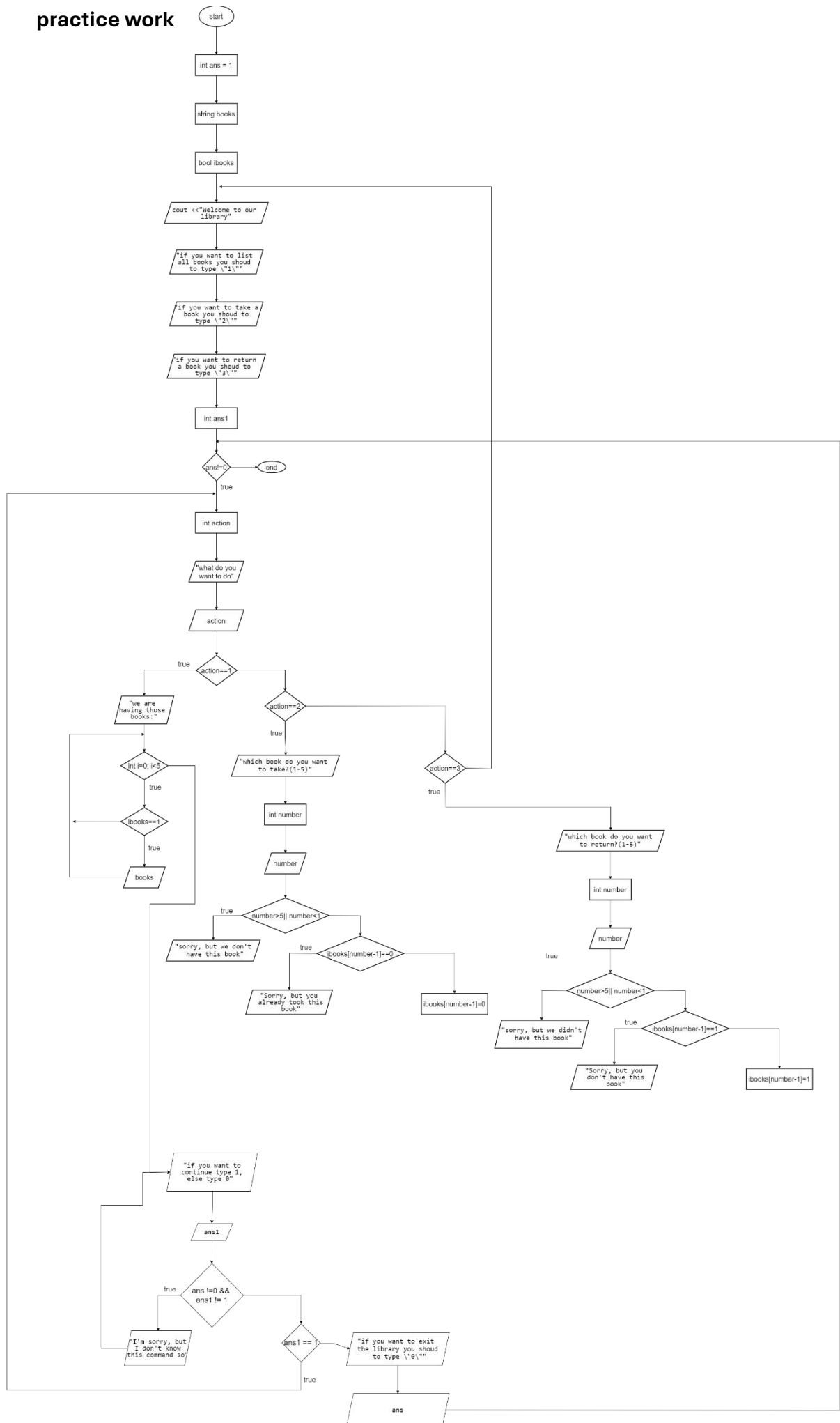
VNS LAB 7 TASK 1



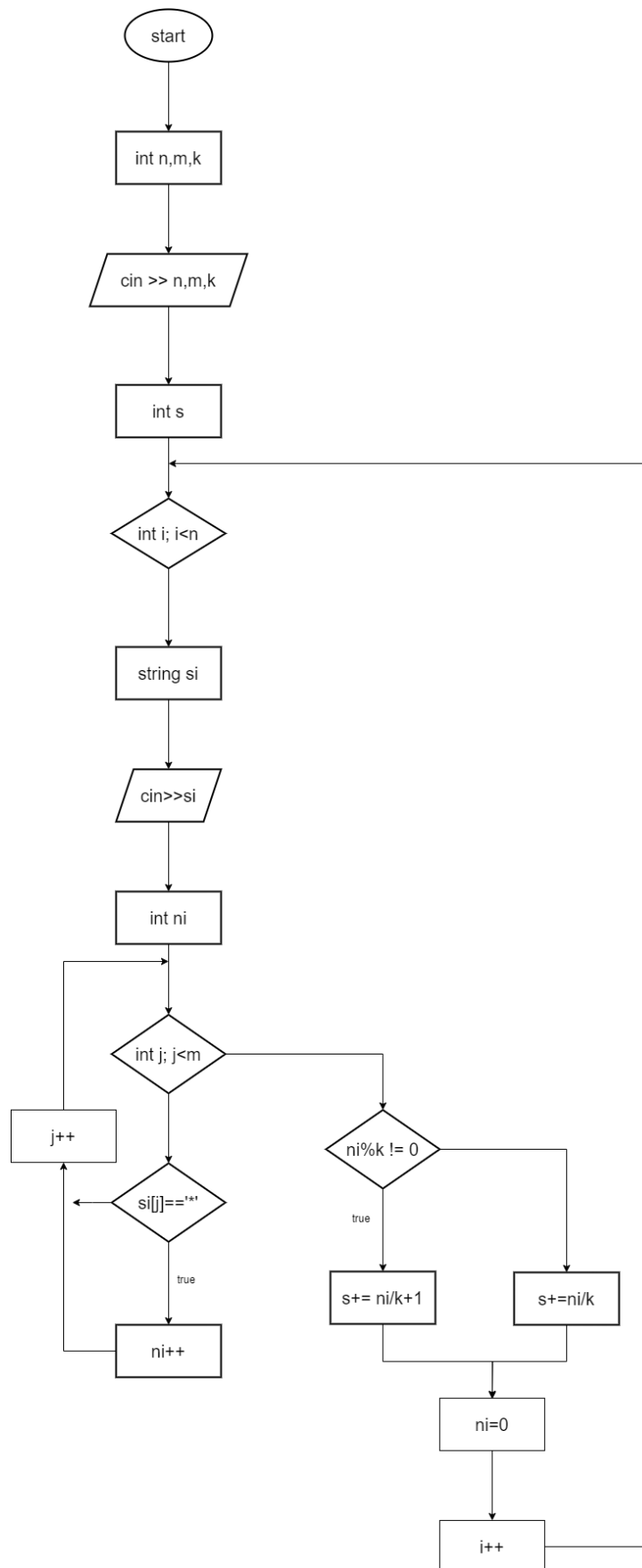
VNS LAB 7 TASK 2



practice work



Self-practice task



Коди:

VNS LAB 2

```
#include <iostream>
```

```
#include <math.h>
```

```
using namespace std;
```

```
double calculation(double b, int n)
```

```
{
```

```
    double a;
```

```
    a=b*pow(n+1,3)/(pow(n,3)*3*n*(3*n-1)*(3*n-2));
```

```
    return a;
```

```
}
```

```
int main()
```

```
{
```

```
    const double e=0.0001;
```

```
    double b=1;
```

```
    int n=1;
```

```
    long double s=b;
```

```
    while (b>e)
```

```
    {
```

```
        b=calculation(b,n);
```

```
        s+=b;
```

```
        n++;
```

```
    }
```

```
    cout<<s<<"\n";
```

```
    return 0;
```

```
}
```

VNS LAB 3

```
#include <stdio.h>
```

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
double count_a(int n, double x)
```

```
{
```

```
    double a = cos(2*n*x)/(4*n*n-1);
```

```
    return a;
```

```
}
```

```
int main()
```

```
{
```

```
    int n=1;
```

```
    double a=0.1;
```

```
    double x=a;
```

```
    double b=0.8;
```

```
    int k=10;
```

```
    double step=(b-a)/k;
```

```
    double sn=0;
```

```
    double se=0;
```

```
    double y;
```

```
    while (x<=b+0.00000001)
```

```
    {
```

```
        while (n<=50)
```

```
        {
```

```
            sn+=count_a(n,x);
```

```
            n++;
```

```

    }

    n=1;

    while (fabs(count_a(n,x))>=0.0001)
    {
        se+=count_a(n,x);

        n++;
    }

    y=0.5-M_PI/4*fabs(sin(x));

    printf("x= %lf \t sn= %lf \t se= %lf \t y= %lf \n", x, sn, se, y);

    x+=step;

    sn=0;

    se=0;

    n=1;
}

return 0;

}

```

VNS LAB 7 TASK 1

```

#include <iostream>

#include <cstdarg>

#include <math.h>


using namespace std;

```

```

int gcd(int a,int b)
{
    int res;
    if (a<b)
    {
        res=a;
    }
    else
    {
        res=b;
    }
    while (res > 1) {
        if (a % res == 0 && b % res == 0)
            break;
        res--;
    }
    return res;
}

```

```

int lcm(int a, int b)
{
    return((a*b)/gcd(a,b));
}

```

```

int nok(int n,...)
{
    va_list a;
    va_start(a,n);

    int res= va_arg(a, int);

```

```

    for (int i=1; i<n;i++)
    {
        res=lmc(res, va_arg(a, int));
    }
    va_end(a);
    return res;
}

```

```

int main()
{
    cout << "lmc of numbers 5,8,9="<<nok(3,5,8,9)<<"\n";
    cout << "lmc of numbers 2,6,9,8,11="<<nok(5,2,6,9,8,11)<<"\n";
    cout << "lmc of numbers 9,17,98,55,88,13="<<nok(6,9,17,98,55,88,13)<<"\n";
    return 0;
}

```

VNS LAB 7 TASK 2

```

#include <iostream>
#include <sstream>

```

```

using namespace std;

```

```

int res(string& str)
{
    istringstream iss(str);
    string word;
    int maxLength = 0;
    while (iss >> word)
    {

```

```
        if (word.length() > maxLength)
        {
            maxLength = word.length();
        }
    }
    return maxLength;
}
```

```
void res(int a[],int n)
{
    int maxim = a[0];
    for (int i = 1; i < n; i++)
    {
        if (a[i] > maxim)
        {
            maxim = a[i];
        }
    }
    cout << "The largest number is " << maxim << endl;
}
```

```
int main()
{

    int n;
    cout << "Enter an amount of numbers: \n";
    cin >> n;
    int a[n];
    cout << "Enter an array of numbers: \n";
    for (int i = 0; i < n; i++)
```



```

{
    cin >> a[i];
}

res(a,n);

string s;

cout << "Enter a string: \n";

cin.ignore();

getline(cin, s);

int longest_word_length = res(s);

cout << "The length of the longest word is: " << longest_word_length << endl;

return 0;
}

```

practice work

```

#include <iostream>

using namespace std;

int main()
{
    int ans=1;

    string books[5]={"Poliana","The C++ Programming Language","Measure, Integration & Real
Analysis","Journey into Discrete Mathematics","English grammar"};

    bool ibooks[5]={1,1,1,1,1};

    menu:

    cout <<"Welcome to our library\n";

    cout <<"if you want to list all books you shoud to type \"1\" \n";

    cout <<"if you want to take a book you shoud to type \"2\" \n";

    cout <<"if you want to return the book you shoud to type \"3\" \n";

```

```

int ans1;
while (ans != 0)
{
    do
    {
        int action;

        cout << "what do you want to do\n";
        cin >> action;
        if (action==1)
        {
            cout << "we are having those books:\n";
            for(int i=0;i<5;i++)
            {
                if (ibooks[i]==1)
                {
                    cout << books[i] << "\n";
                }
            }
        }
        else if (action==2)
        {
            cout << "which book do you want to take?(1-5)\n";
            int number;
            cin >> number;
            if (number>5 || number<1)
            {
                cout << " sorry, but we don't have this book\n";
            }
            else
            {

```

```
        if (ibooks[number-1]==0)
        {
            cout << "Sorry, but you already took this book\n";
        }
        else
        {
            ibooks[number-1]=0;
        }
    }
}
else if (action==3)
{
    cout << "which book do you want to return the book?(1-5)\n";
    int number;
    cin >> number;
    if (number>5 || number<1)
    {
        cout << " sorry, but we didn't have this book\n";
    }
    else
    {
        if (ibooks[number-1]==1)
        {
            cout << "Sorry, but you don't have this book\n";
        }
        else
        {
            ibooks[number-1]=1;
        }
    }
}
```

```

    }

    else

    {

        goto menu;

    }

    question:

    cout<<"if you want to continue type 1, else type 0 \n";

    cin>>ans1;

    if (ans1 != 1 && ans1 != 0)

    {

        cout << "I'm sorry, but I don't know this command\n so ";

        goto question;

    }

    }while (ans1==1);

    cout <<"if you want to exit the library you should to type \"0\" \n";

    cin >> ans;

};

return 0;

}

```

Self-practice task

```

#include <iostream>


using namespace std;


int main()

{

    int n, m, k;

    cin >> n >> m >> k;

```

```
int s = 0;
```

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

```
{
```

```
    string si;
```

```
    cin >> si;
```

```
    int ni=0;
```

```
    for (int j=0;j<m;j++)
```

```
    {
```

```
        if (si[j]=='*')
```

```
        {
```

```
            ni++;
```

```
        }
```

```
    }
```

```
    if (ni % k != 0)
```

```
    {
```

```
        s += ni / k + 1;
```

```
    }
```

```
    else
```

```
    {
```

```
        s += ni / k;
```

```
    }
```

```
    ni=0;
```

```
}
```

```
cout << s << endl;
```

```
return 0;
```

```
}
```

Тестування

VNS LAB 2

```
584 \mingw64\bin\gdb.exe --interpreter=mi  
2.37101  
PS C:\Users\TheMo\OneDrive\Desktop\nauka>
```

VNS LAB 3

x= 0.100000	sn= 0.421368	se= 0.419761	y= 0.421591
x= 0.170000	sn= 0.366845	se= 0.363937	y= 0.367124
x= 0.240000	sn= 0.313139	se= 0.312368	y= 0.313309
x= 0.310000	sn= 0.260381	se= 0.261161	y= 0.260408
x= 0.380000	sn= 0.208758	se= 0.208049	y= 0.208680
x= 0.450000	sn= 0.158490	se= 0.158041	y= 0.158379
x= 0.520000	sn= 0.109830	se= 0.109454	y= 0.109751
x= 0.590000	sn= 0.063046	se= 0.064318	y= 0.063035
x= 0.660000	sn= 0.018410	se= 0.018718	y= 0.018459
x= 0.730000	sn= -0.023831	se= -0.023042	y= -0.023758
x= 0.800000	sn= -0.063464	se= -0.063379	y= -0.063410

VNS LAB 7 TASK 1

```
584 \mingw64\bin\gdb.exe --interpreter=mi  
lmc of numbers 5,8,9=360  
lmc of numbers 2,6,9,8,11=792  
lmc of numbers 9,17,98,55,88,13=42882840
```

VNS LAB 7 TASK 2

```
Enter an amount of numbers:  
7  
Enter an array of numbers:  
8 1 5 7 9 3 2  
The largest number is 9  
Enter a string:  
Stanko Oleh's program  
The length of the longest word is: 7
```

Робота в команді:

Ми з командою збиралися разом 1 раз для обговорення 3 епіка.



Посилання на файли у пулл реквесті

VNS LAB 2

[epic 3 - Oleh Stanko by Stankoleh · Pull Request #182 · artificial-intelligence-department/ai_programming_playground_2024](#)

VNS LAB 3

[epic 3 - Oleh Stanko by Stankoleh · Pull Request #182 · artificial-intelligence-department/ai_programming_playground_2024](#)

VNS LAB 7 TASK 1

[epic 3 - Oleh Stanko by Stankoleh · Pull Request #182 · artificial-intelligence-department/ai_programming_playground_2024](#)

VNS LAB 7 TASK 2

[epic 3 - Oleh Stanko by Stankoleh · Pull Request #182 · artificial-intelligence-department/ai_programming_playground_2024](#)

Висновок: я навчився використовувати перезавантажені функції та покращив роботу з циклами й їх застосуваннями. Також я ознайомився з базовими математичними функціями