

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

Мартин Максим Ігорович

## **Тема:**

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

## **Мета:**

Вдосконалити використання відповідних циклів до певних ситуацій.

Створення програми, побудованої на введенні даних, поки користувач не захоче вийти.

Розібратись з перевантаженими функціями та як вони використовуються.

Розібратись як використовують глобальні змінні та як їх створювати.

## **Теоретичні відомості:**

1) Вивчив/знав:

1. Цикли та їх види
2. Робота з циклами
3. Створення та виклик функцій
4. Робота з функціями

2) Джерела:

Всю інформацію до теоретичних відомостей я отримав на лекційних/практичних парах. Додатково використовував сайт

<https://acode.com.ua/>

## **Виконання роботи:**

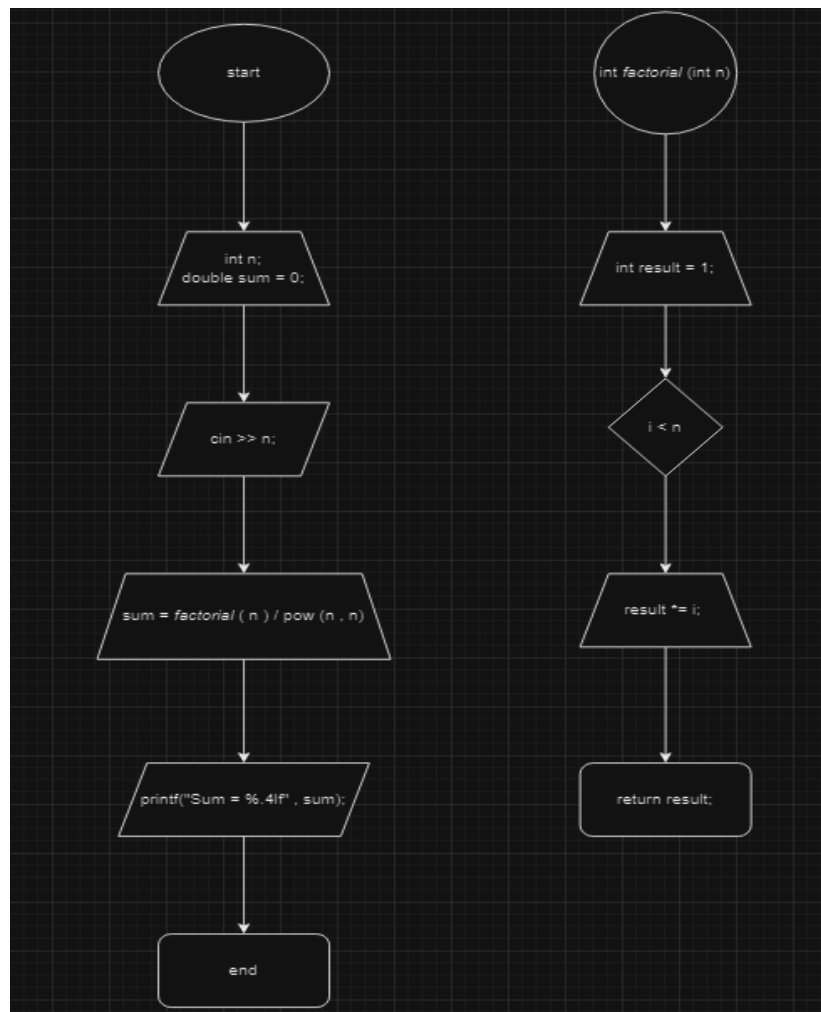
### **Завдання №1 - VNS Lab 2 , Варіант №11**

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

$$a_n = \frac{n!}{n^n}$$

Блок схема , код та тестування:



```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int factorial (int n){
6      int result = 1;
7      for (int i = 2; i <= n; i++)
8      {
9          result *= i;
10     }
11     return result;
12 }
13
14 int main(){
15     int n;
16     double sum = 0;
17     cout << "n = ";
18     cin >> n;
19     sum = factorial(n)/pow(n , n);
20     printf("Sum = %.4lf" , sum);
21
22     return 0;
23 }

```

```

n = 2
Sum = 0.5000

```

## Завдання №2 - VNS Lab 3 , Варіант №11

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  $\varepsilon$  ( $\varepsilon=0.0001$ ).

Для порівняння знайти точне значення функції.

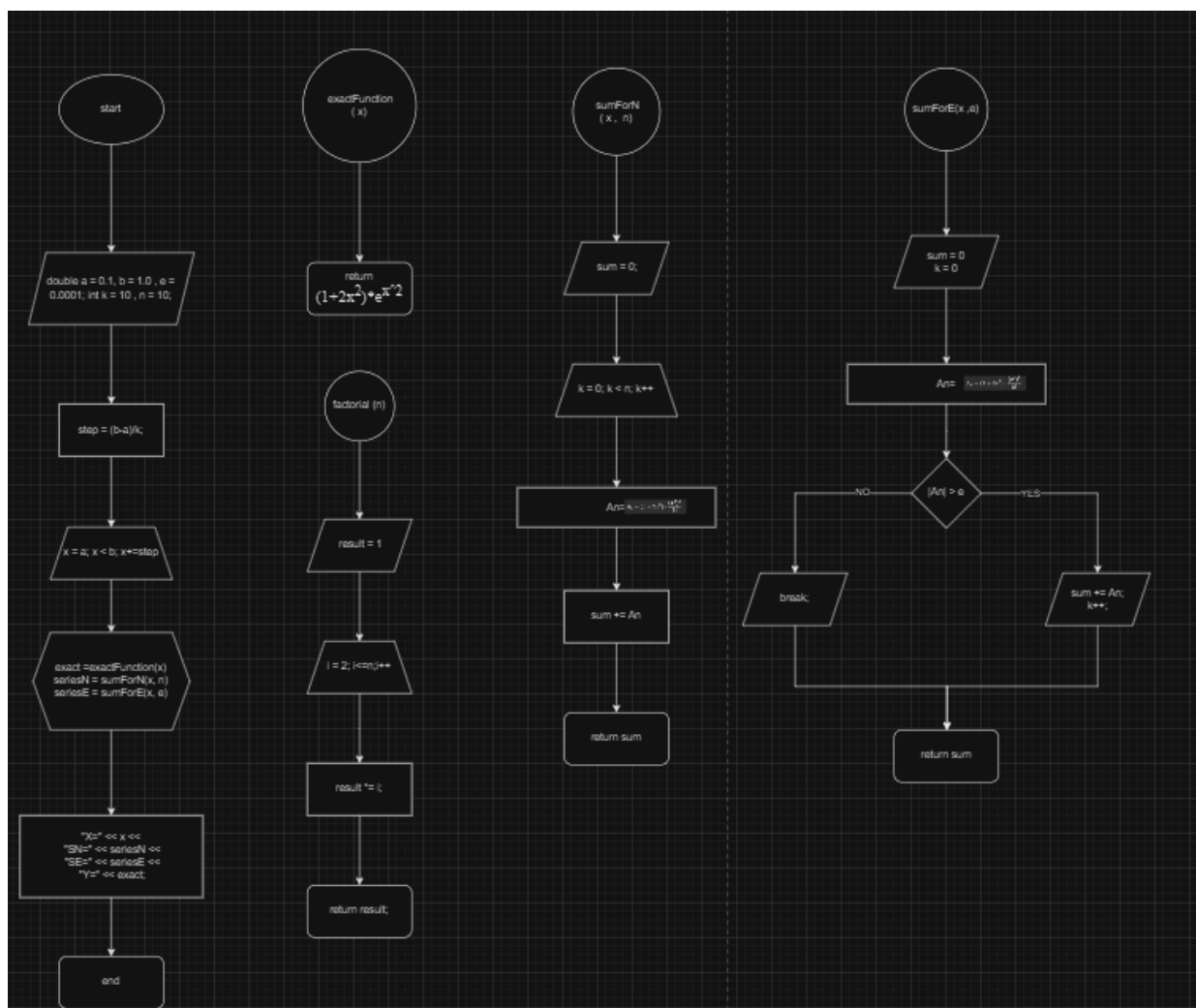
Функція:  $y = (1+2x^2)*e^{x^2}$

Діапазон зміни аргументу:  $0,1 \leq x \leq 1$

$n = 10$

сума :  $S = 1+3x^2+\dots+\frac{2n+1}{n!}*x^{2n}$

Блок схема , код та тестування:



```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  double exactFunction(double x){
6  |   return (1 + 2 * pow(x, 2)) * exp(pow(x, 2));
7  }
8
9  int factorial (int n){
10 |   int result = 1;
11 |   for (int i = 2; i <= n; i++)
12 |   {
13 |       result *= i;
14 |   }
15 |   return result;
16 }
17
18 // Функція для обчислення значення ряду для заданого числа членів n
19 double sumForN(double x , int n){
20 |   double sum = 0;
21 |   for (int k = 0; k < n; k++)
22 |   {
23 |       double An = (1 + 2 * pow(x, 2)) * pow(pow(x, 2), k) / factorial(k);
24 |       sum += An;
25 |   }
26 |   return sum;
27 }

```

```

29 // Функція для обчислення значення ряду для заданої точності e
30 //Цей цикл додає члени ряду, поки вони значно впливають на результат.
31 double sumForE(double x ,double e){
32 |   double sum = 0;
33 |   double An;
34 |   int k = 0;
35 |   while (true){
36 |       An = (1 + 2 * pow(x, 2)) * pow(pow(x, 2), k) / factorial(k); // Степеневий Ряд Тейлора  $e^{(x^2)} = 1 + (x^2/1!) + (x^4/2!) + \dots$ 
37 |       if (abs(An) > e){ // abs - модуль , використовується для перевірки абсолютного значення змінної An
38 |           sum += An;
39 |           k++;
40 |       }
41 |       else{
42 |           break;
43 |       }
44 |       return sum;
45 }

```

```

48 int main(){
49 |   double a = 0.1, b = 1.0 , e = 0.0001;
50 |   int k = 10 , n = 10;
51 |   double step = (b-a)/k;
52
53 |   for (double x = a; x < b; x+=step){
54 |       double exact = exactFunction(x);
55 |       double seriesN = sumForN(x, n);
56 |       double seriesE = sumForE(x, e);
57
58 |       cout << "x=" << x << "\tSN=" << seriesN << "\tSE=" << seriesE << "\tY=" << exact << endl;
59 |       //Тут X- значення параметра; SN- значення суми для заданого n; SE- значення суми для заданої точності; Y-точне значення функції.
60 |   }
61 |   return 0;
62 }

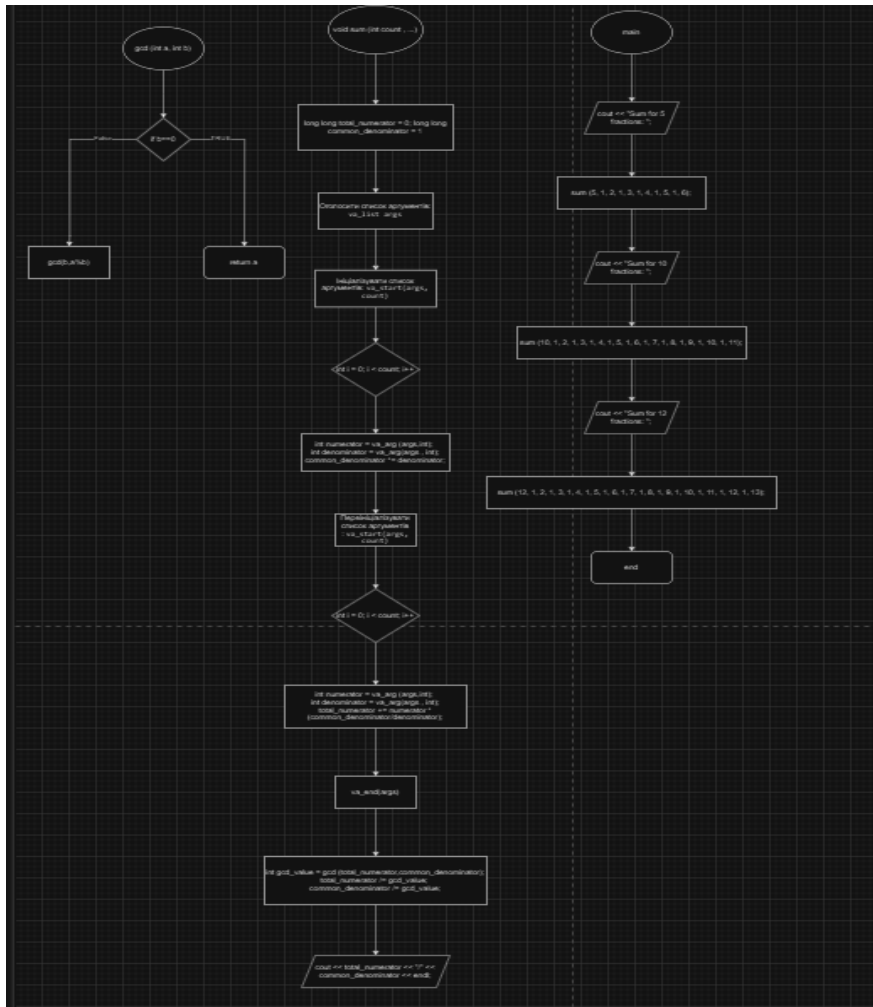
```

X=0.1	SN=1.03025	SE=1.02	Y=1.03025
X=0.19	SN=1.11161	SE=1.0722	Y=1.11161
X=0.19	SN=1.11161	SE=1.0722	Y=1.11161
X=0.28	SN=1.25114	SE=1.1568	Y=1.25114
X=0.37	SN=1.46068	SE=1.2738	Y=1.46068
X=0.46	SN=1.75858	SE=1.4232	Y=1.75858
X=0.55	SN=2.17195	SE=1.605	Y=2.17195
X=0.64	SN=2.74011	SE=1.8192	Y=2.74011
X=0.73	SN=3.51985	SE=2.0658	Y=3.51985
X=0.82	SN=4.59331	SE=2.3448	Y=4.59331
X=0.91	SN=6.07995	SE=2.6562	Y=6.07995
X=1	SN=8.15484	SE=3	Y=8.15485

### Завдання №3 - VNS Lab 3 , Варіант №11 , Завдання 1

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. Розв'язати завдання, використовуючи функції зі змінною кількістю параметрів.

Блок схема , код та тестування:



```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4
5 // Пошук найбільшого спільного дільника (використ. рекурс. функ.)
6 int gcd (int a, int b){
7     if (b==0){
8         return a;
9     }
10    else {
11        return gcd(b,a%b);
12    }
13 }
```

```

15 void sum (int count , ...){
16     long long total_numerator = 0;
17     long long common_denominator = 1;
18
19     va_list args;          // Для змінної кількості параметрів
20     va_start(args,count); // зберігає адреси всіх аргументів після коунт
21
22     for (int i = 0; i < count; i++){
23         int numerator = va_arg (args,int);
24         int denominator = va_arg(args , int);
25         common_denominator *= denominator; // <--
26     }
27     va_start (args, count); // restart
28
29     for (int i = 0; i< count;i++){
30         int numerator = va_arg (args,int);
31         int denominator = va_arg(args , int);
32         total_numerator += numerator * (common_denominator/denominator);
33     }
34     va_end(args);
35
36     int gcd_value = gcd (total_numerator,common_denominator);
37     total_numerator /= gcd_value;
38     common_denominator /= gcd_value;
39     cout << total_numerator << "/" << common_denominator << endl;
40 }
41
42 int main (){
43     cout << "Sum for 5 fractions: ";
44     sum (5, 1, 2, 1, 3, 1, 4, 1, 5, 1, 6);
45
46     cout << "Sum for 10 fractions: ";
47     sum(10, 1, 2, 1, 3, 1, 4, 1, 5, 1, 6, 1, 7, 1, 8, 1, 9, 1, 10, 1, 11);
48
49     cout << "Sum for 12 fractions: ";
50     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);
51
52     return 0;
53 }
54 // використав Цикли, Функції, Функції зі змінною кількістю параметрів (еліпсис), Рекурсія та Вбудовані функції

```

```

Sum for 5 fractions: 29/20
Sum for 10 fractions: 55991/27720
Sum for 12 fractions: 106060455/48648600

```

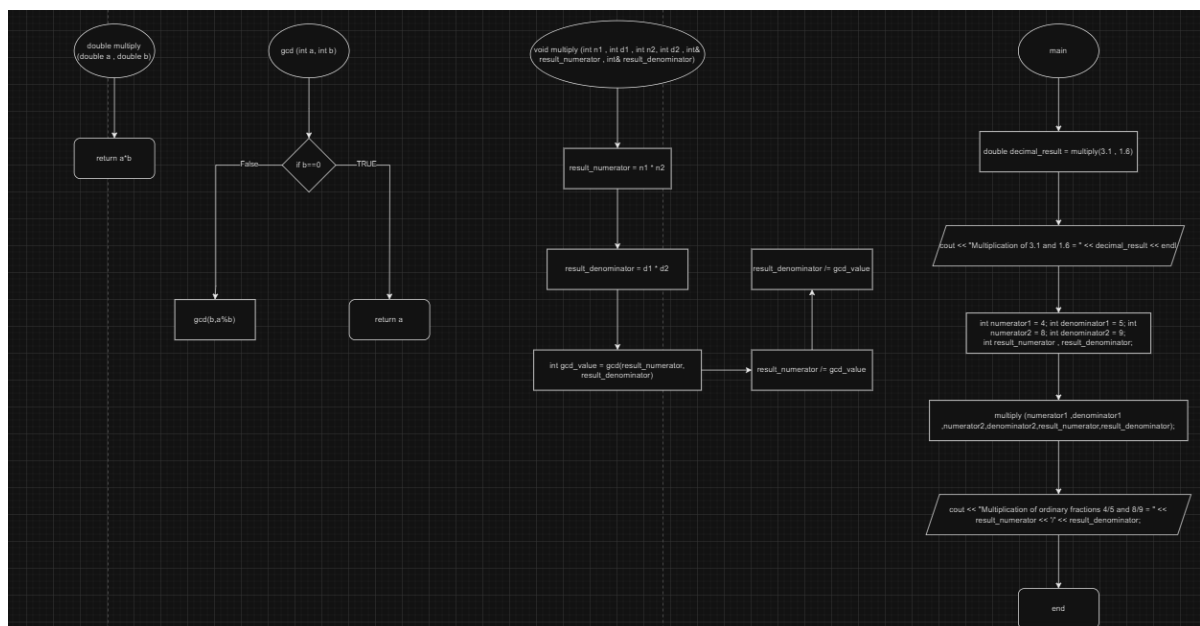
## Завдання №4 - VNS Lab 3 , Варіант №11 , Завдання 2

Написати перевантажені функції й основну програму, що їх викликає.

а) для множення десяткових дробів;

б) для множення звичайних дробів.

Блок схема , код та тестування:



```

1  #include <iostream>
2  using namespace std;
3
4  double multiply (double a , double b){
5      return a*b;
6  }
7
8  int gcd (int a, int b){
9      if (b==0){
10         return a;
11     }
12     else {
13         return gcd(b,a%b);
14     }
15 }
16
17 void multiply (int n1 , int d1 , int n2, int d2 , int& result_numerator , int& result_denominator){
18     result_numerator = n1 * n2;
19     result_denominator = d1 * d2;
20
21     int gcd_value = gcd(result_numerator, result_denominator);
22     result_numerator /= gcd_value;
23     result_denominator /= gcd_value;
24 }
25 int main (){
26     double decimal_result = multiply(3.1 , 1.6);
27     cout << "Multiplication of 3.1 and 1.6 = " << decimal_result << endl;
28
29     int numerator1 = 4; int denominator1 = 5;
30     int numerator2 = 8; int denominator2 = 9;
31
32     int result_numerator , result_denominator;
33     multiply (numerator1 ,denominator1 ,numerator2,denominator2,result_numerator,result_denominator);
34     cout << "Multiplication of ordinary fractions 4/5 and 8/9 = " << result_numerator << '/' << result_denominator;
35     return 0;
36 }
37 //використав перевантаження функції та рекурсію

```

Multiplication of 3.1 and 1.6 = 4.96

Multiplication of ordinary fractions 4/5 and 8/9 = 32/45

## Завдання №5 – Practice work

Ви розробляєте просту програму для керування бібліотекою. У бібліотеці вже є певна кількість книг, які користувачі можуть позичати або повертати.

### Функціональні можливості програми:

- Вивести список усіх книг.
- Дозволити взяти книгу (за умови наявності).
- Дозволити повернути книгу.

### Структури даних:

- Використовуйте масив або вектор для зберігання назв книг.
- Використовуйте інший масив або вектор для збереження стану доступності кожної книги.

**Мета завдання:** Навчитися використовувати оператори циклів та функцію переходу на мітку:

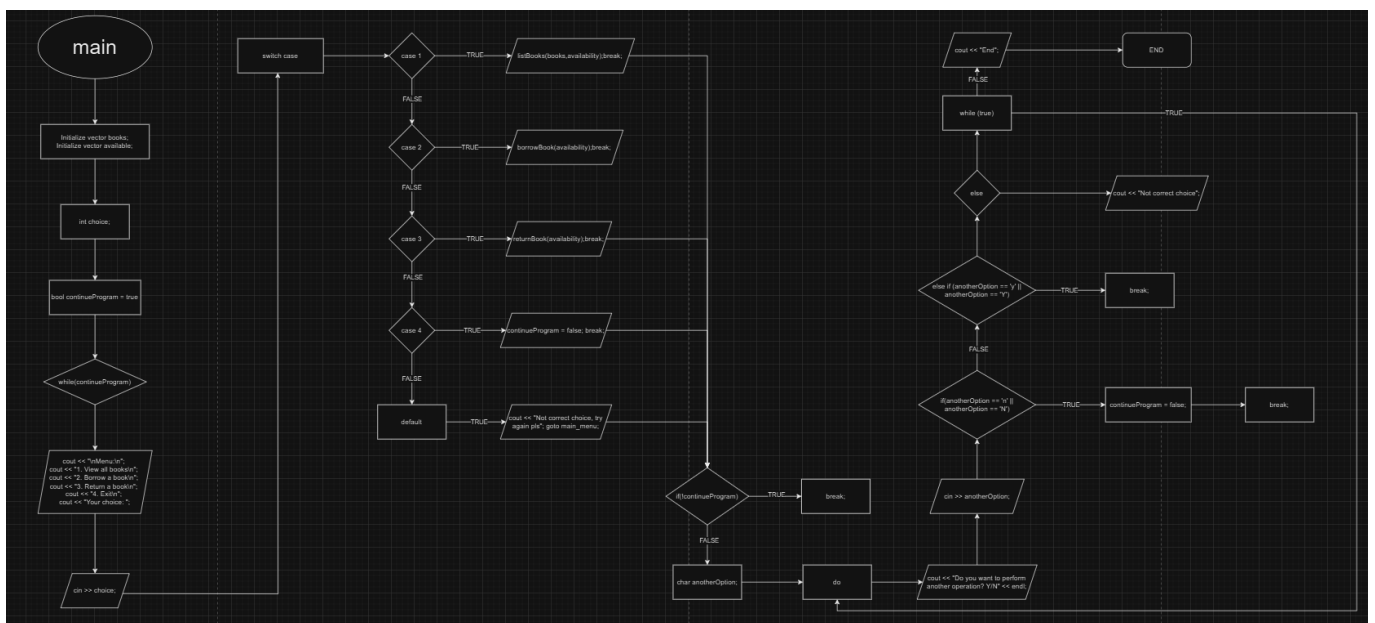
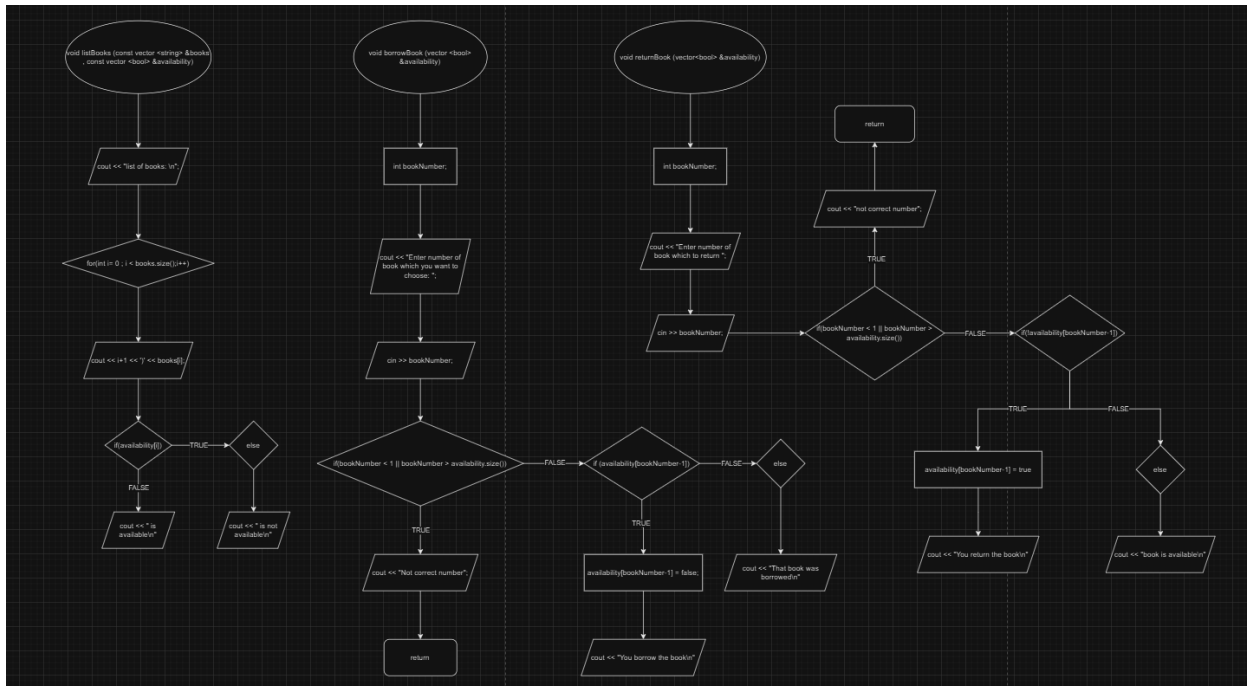
- for() { ... }
- for each
- while() { ... }
- do { ... } while()
- goto



## Вимоги до програми:

- **while:** програма повинна працювати доти, доки користувач не вирішить завершити роботу.
- **do while:** після кожної операції (позичити, повернути, вивести список) запитуйте користувача, чи хоче він виконати ще одну операцію. Якщо так, програма повертається до меню.
- **for:** використовуйте для виведення списку всіх книг.
- **for each:** використовуйте для перевірки наявності кожної книги.
- **goto:** у разі неправильного вибору користувача використовуйте goto, щоб повернути його до головного меню.

Блок схема , код та тестування:



```

1  #include <iostream>
2  #include <vector>
3  #include <string>
4  using namespace std;
5
6  void listBooks (const vector <string> &books , const vector <bool> &availability){
7      cout << "list of books: \n";
8      for(int i= 0 ; i < books.size();i++){
9          cout << i+1 << ' ' << books[i];
10         if(availability[i]){
11             cout << " is available\n";
12         } else {
13             cout << " is not available\n";
14         }
15     }
16 }
17
18 void borrowBook (vector <bool> &availability){
19     int bookNumber;
20     cout << "Enter number of book which you want to choose: ";
21     cin >> bookNumber;
22
23     if(bookNumber < 1 || bookNumber > availability.size()){
24         cout << "Not correct number";
25         return;
26     }
27
28     if (availability[bookNumber-1]){
29         availability[bookNumber-1] = false;
30         cout << "You borrow the book\n";
31     } else {
32         cout << "That book was borrowed\n";
33     }
34 }

```

```

35
36 void returnBook (vector<bool> &availability){
37     int bookNumber;
38     cout << "Enter number of book which to return ";
39     cin >> bookNumber;
40     if(bookNumber < 1 || bookNumber > availability.size()){
41         cout << "not correct number";
42         return;
43     }
44     if(!availability[bookNumber-1]){
45         availability[bookNumber-1] = true;
46         cout << "You return the book\n";
47     } else {
48         cout << "book is available\n";
49     }
50 }
51
52 int main(){
53     vector <string> books = {"book 1" , "book 2" , "book 3" , "book 4"};
54     vector <bool> availability(books.size(),true);
55
56     int choice;
57     bool continueProgram = true;
58
59     while(continueProgram){
60         main_menu:
61
62         cout << "\nMenu:\n";
63         cout << "1. View all books\n";
64         cout << "2. Borrow a book\n";
65         cout << "3. Return a book\n";
66         cout << "4. Exit\n";
67         cout << "Your choice: ";
68         cin >> choice;
69

```

```

70     switch (choice) {
71     case 1:
72         listBooks(books,availability);
73         break;
74     case 2:
75         borrowBook(availability);
76         break;
77     case 3:
78         returnBook(availability);
79         break;
80     case 4:
81         continueProgram = false;
82         break;
83     default:
84         cout << "Not correct choice, try again pls";
85         goto main_menu;
86     }
87
88     if(!continueProgram)
89         break;
90
91     char anotherOption;
92     do {
93         cout << "Do you want to perform another operation? Y/N" << endl;
94         cin >> anotherOption;
95         if(anotherOption == 'n' || anotherOption == 'N'){
96             continueProgram = false;
97             break;
98         } else if (anotherOption == 'y' || anotherOption == 'Y'){
99             break;
100         } else {
101             cout << "Not correct choice";
102         }
103     } while (true);
104 }
105
106 cout << "End";
107 return 0;
108 }

```

```

Menu:
1. View all books
2. Borrow a book
3. Return a book
4. Exit
Your choice: 1
list of books:
1)book 1 is available
2)book 2 is available
3)book 3 is available
4)book 4 is available
Do you want to perform another operation? Y/N
Y

Menu:
1. View all books
2. Borrow a book
3. Return a book
4. Exit
Your choice: 2
Enter number of book which you want to choose: 1
You borrow the book
Do you want to perform another operation? Y/N
y

Menu:
1. View all books
2. Borrow a book
3. Return a book
4. Exit
Your choice: 3
Enter number of book which to return 4
book is available
Do you want to perform another operation? Y/N
N
End

```

## Завдання №6 – Self practice algotester

### Лотарея

Одного разу двоє друзів, Віталік та Роман, вирішили зіграти в лотерею і навіть купили відповідний білет. На лотерейному білеті є прямокутна таблиця розміром  $n \times m$ . У кожній клітинці таблиці записане одне ціле число. Для участі в лотереї необхідно замалювати рівно одне число з таблиці та відіслати білет організаторам. Віталік переконаний, що необхідно обрати найменше число, проте Роман абсолютно впевнений, що переможе найбільше. Білет у хлопців лише один, і вони довго не могли вирішити, як їм учинити. Після декількох днів активних суперечок та наукових дискусій на тему «Чому малі числа кращі, ніж великі» чи навпаки, друзі вирішили зробити так: спочатку Віталік обирає стовпець, а тоді Роман вибирає число з цього стовпця.

Ваше завдання визначити, яке число все-таки оберуть хлопці.

### Вхідні дані

У першому рядку два цілі числа  $n$  та  $m$  — кількість рядків та стовпців лотерейної таблиці.

У наступних  $n$  рядках по  $m$  цілих чисел  $a_{ij}$  —  $j$ -те число в  $i$ -му рядку лотерейної таблиці.

### Вихідні дані

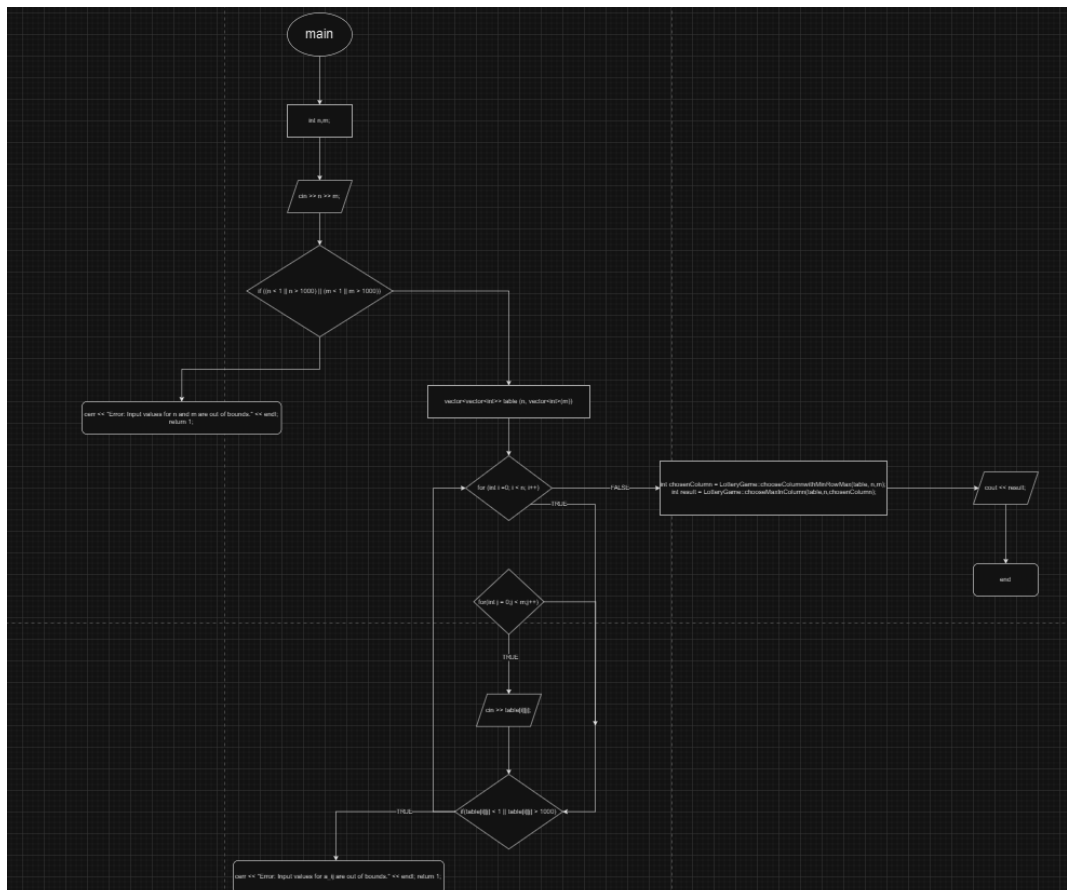
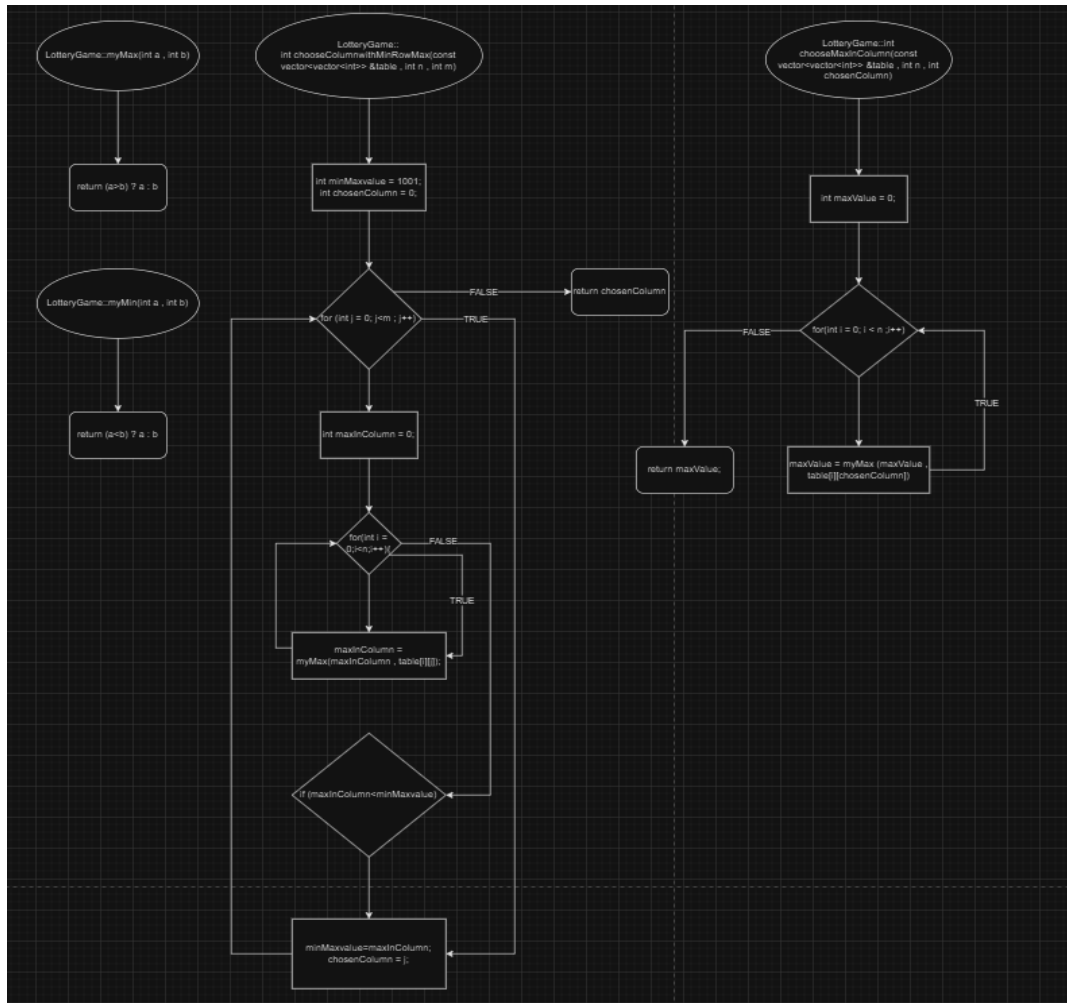
У єдиному рядку виведіть число, яке виберуть Віталік та Роман.

### Обмеження

40% тестів:  $1 \leq n, m \leq 50$ ,  $1 \leq a_{ij} \leq 10^3$ .

60% тестів:  $51 \leq n, m \leq 10^3$ ,  $1 \leq a_{ij} \leq 10^3$ .

Блок схема, код та тестування:



```

1 //https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/25#mySolutions
2 #include <iostream>
3 #include <vector>
4 using namespace std;
5
6 namespace LotteryGame {
7     int myMax(int a,int b){
8         return (a>b) ? a : b;
9     }
10
11     int myMin(int a , int b){
12         return (a<b) ? a : b;
13     }
14     int chooseColumnwithMinRowMax(const vector<vector<int>> &table , int n , int m){
15         int minMaxvalue = 1001;
16         int chosenColumn = 0;
17         for (int j = 0; j<m ; j++){
18             int maxInColumn = 0;
19             for(int i = 0;i<n;i++){
20                 maxInColumn = myMax(maxInColumn , table[i][j]);
21             }
22             if (maxInColumn<minMaxvalue){
23                 minMaxvalue=maxInColumn;
24                 chosenColumn = j;
25             }
26         }
27         return chosenColumn;
28     }
29     int chooseMaxInColumn(const vector<vector<int>> &table , int n , int chosenColumn){
30         int maxValue = 0;
31         for(int i = 0; i < n ;i++){
32             maxValue = myMax (maxValue , table[i][chosenColumn]);
33         }
34         return maxValue;
35     }
36 }
37
38 int main (){
39     int n,m;
40     cin >> n >> m;
41     if ((n < 1 || n > 1000) || (m < 1 || m > 1000)) {
42         cerr << "Error: Input values for n and m are out of bounds." << endl;
43         return 1;
44     }
45
46     vector<vector<int>> table (n, vector<int>(m));
47
48     for (int i =0; i < n; i++){
49         for(int j = 0;j < m;j++){
50             cin >> table[i][j];
51             if(table[i][j] < 1 || table[i][j] > 1000){
52                 cerr << "Error: Input values for a_ij are out of bounds." << endl;
53                 return 1;
54             }
55         }
56     }
57     int chosenColumn = LotteryGame::chooseColumnwithMinRowMax(table, n,m);
58     int result = LotteryGame::chooseMaxInColumn(table,n,chosenColumn);
59     cout << result;
60     return 0;
61 }

```

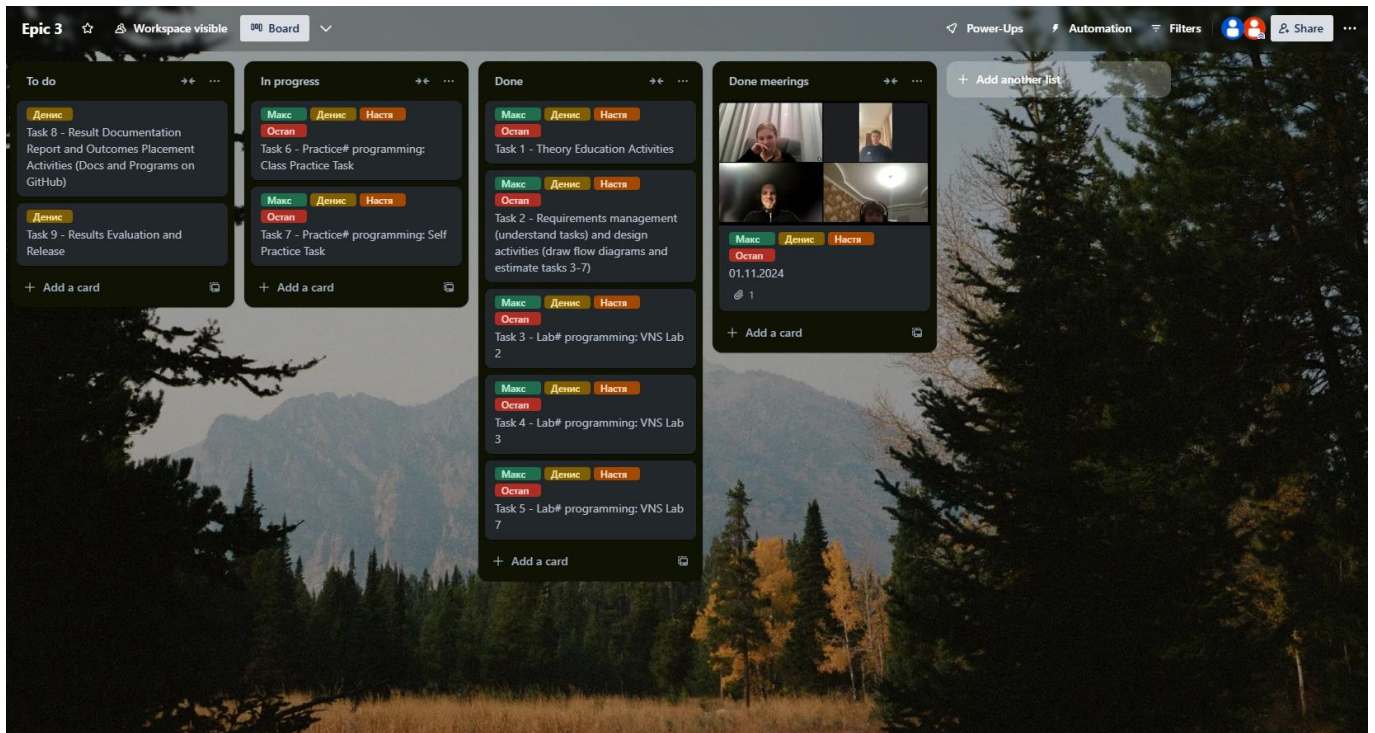
Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.179	4.352	<a href="#">Перегляд</a>

```

3 3
1 2 7
18 25 99
24 24 24
24

```

## Робота з командою:



## Висновок:

Вдосконали використання відповідних циклів до конкретних варіантів ситуацій,розібрався як використовувати глобальні змінні , використав математичні формули для функцій щоб вирішити задачі в коді навчився/знав як користуватися операторами циклів та функцією переходу на мітку.