

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 ІІІ-11

Корнілов Артем Сергійович

Львів 2024

## Тема:

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

## Мета:

Розібратись з перезавантаженими функціями та як їх використовувати. Покращити роботу з циклами та їх застосуваннями.

## Теоретичні відомості:

1. Введення в Цикли та їх Види в C++:
2. Управління Виконанням Циклів:
3. Вкладені Цикли:

### Джерела

4. Основи Функцій у C++:
5. Перевантаження Функцій та Простір Імен:
6. Розширені Можливості Функцій:
7. Вбудовані Функції в C++:

### Джерела

- 1)[C++ • Теорія • Урок 48 • Перезавантаження функцій](#)
- 2)[C++ • Теорія • Урок 43 • Функції](#)
- 3)[C++ • Теорія • Урок 44 • Область видимості, глобальні/локальні дані, static/extern](#)
- 4)[C++ • Теорія • Урок 47 • inline функції](#)

## Виконання роботи:

1) *Опрацювання завдання та вимог до програми та середовища*

### Завдання №1 Class Practice Task

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.

### Вимоги:

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 №2 VNS Lab 2 - Task 1-25

Завдання:.

25) Знайти суму 7 членів ряду, у якому

$$a_n = n^2 e^{-\sqrt{n}}$$

### Вимоги:

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

## Завдання №3 VNS Lab 3 - Task 1-25

Завдання:

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.

### Умови:

$y = \frac{e^x - e^{-x}}{2}$	$0,1 \leq x \leq 1$	20	$S = x + \frac{x^3}{3!} + \dots + \frac{x^{2n+1}}{(2n+1)!}$
------------------------------	---------------------	----	---

## Завдання №4 VNS Lab 7 - Task 1-25

Завдання:

Написати функцію (або макровизначення), що знаходить довжину сторони по координатах його точок.. Написати функцію square, що обчислює площу трикутника, заданого координатами вершин.

Написати функцію square1 з змінною кількістю параметрів, що визначає площу трикутника, що містить діагональ найбільшої довжини опуклого багатокутника, заданого координатами своїх вершин

### Умови:

## **Завдання №5 VNS Lab 7 - Task 2-25**

**Завдання:**

Написати перевантажені функції й основну програму, що їх викликає.

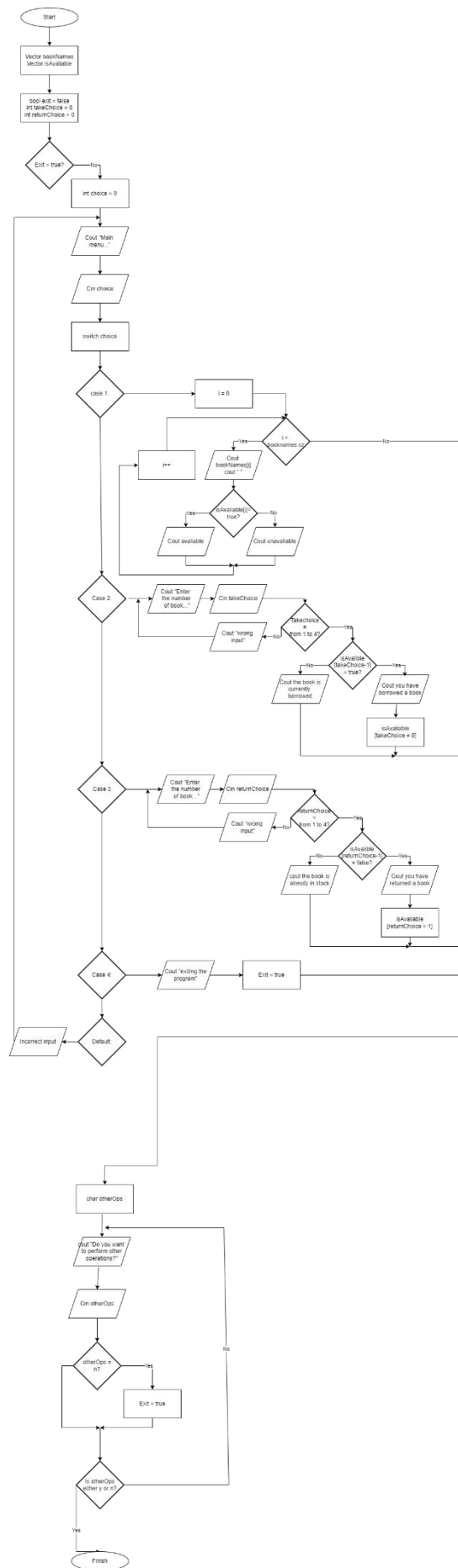
**Умови:**

а) для двовимірного масиву знищує всі парні рядки; б) для одновимірного масиву знищує всі елементи, що містяться між двома нульовими елементами.

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

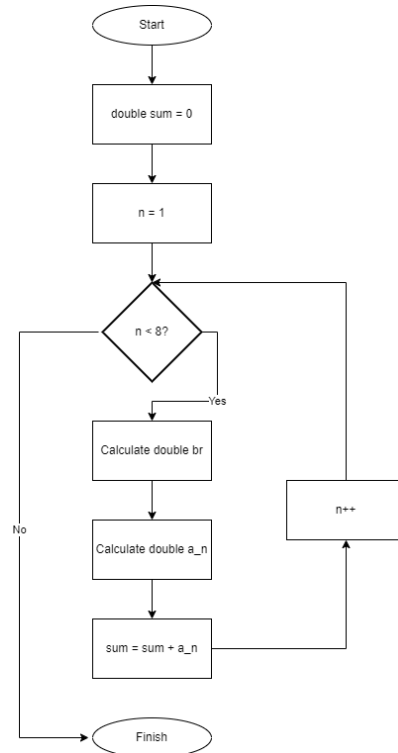
**Завдання №1:** Планований час виконання 2години.

**Class Practice Task**



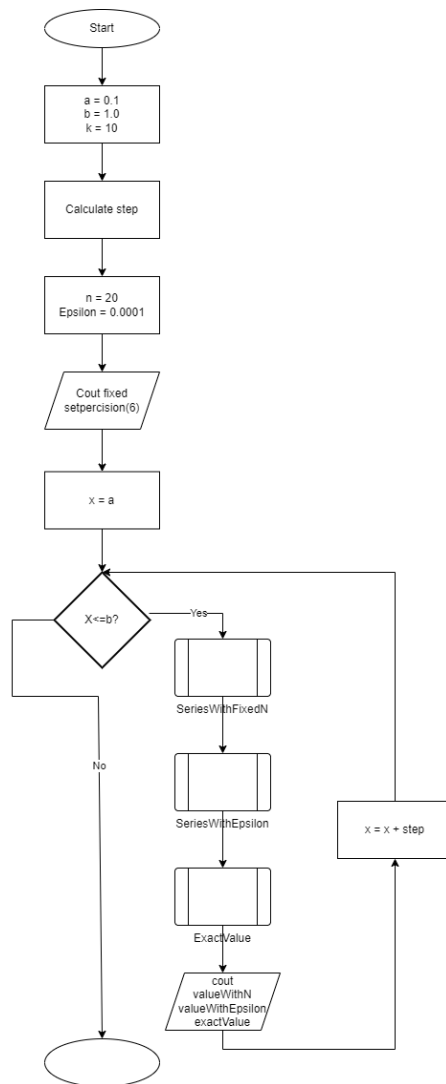
**Завдання №2:** Планований час виконання 10хвилин.

**VNS Lab 2 - Task 1-1**

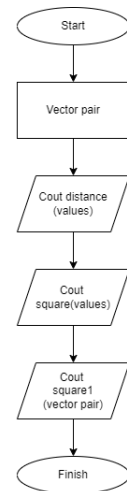
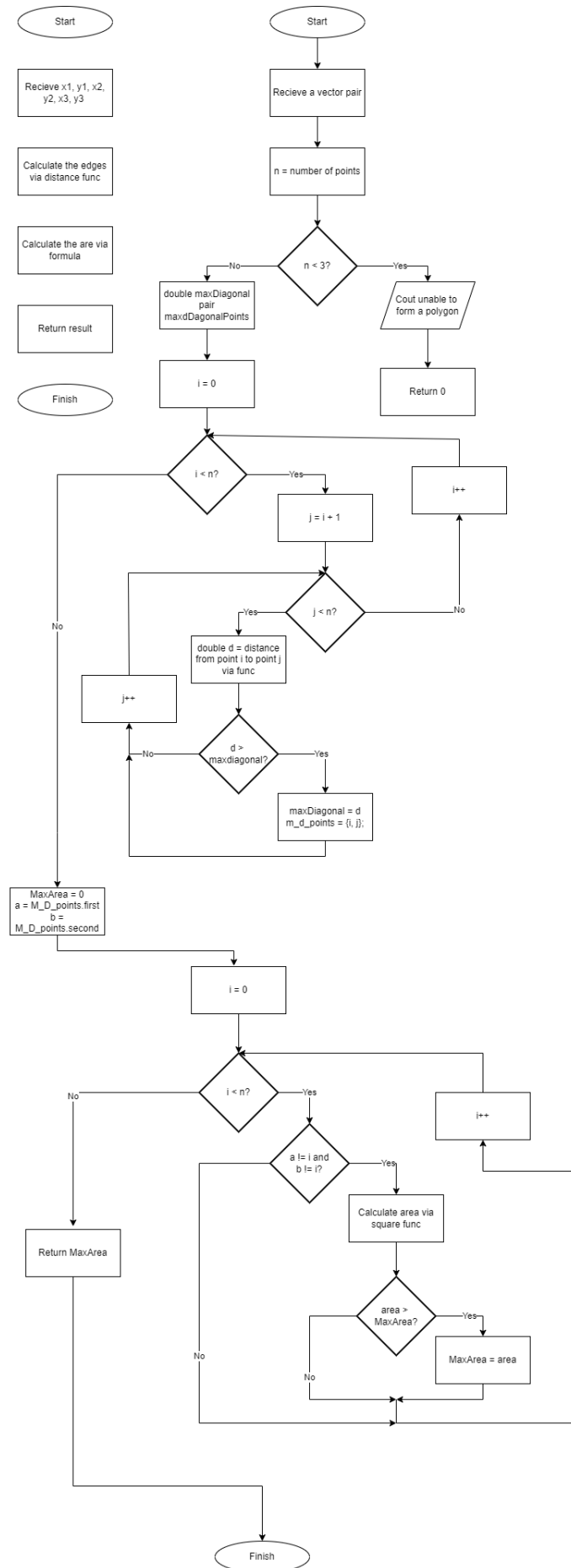
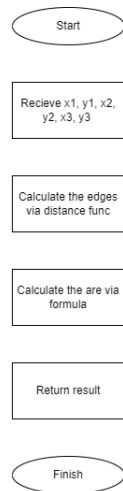
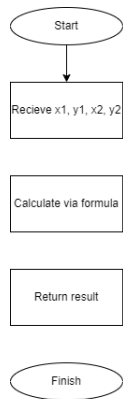


**Завдання №3:** Планований час виконання 40хвилин.

**VNS Lab 3 - Task 1-1**

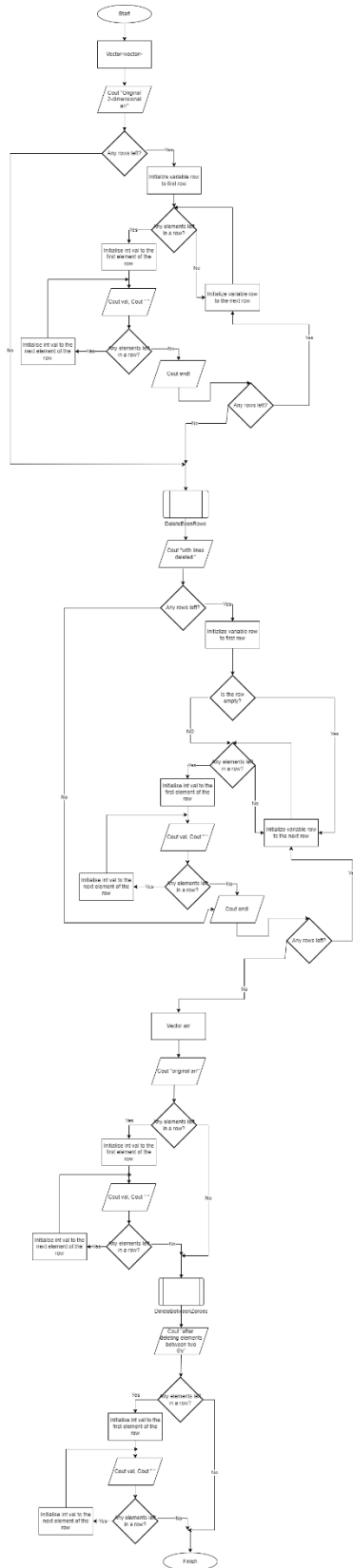


**Завдання №4:** Планований час виконання 20хвилин.  
**VNS Lab 7 - Task 1-1**





## VNS Lab 7 - Task 2-1



## Код програм з посиланням на зовнішні ресурси:

### Завдання №1:

#### [Class Practice Task](#)

```
code > practice_work_task_1_arten_kornilov.cpp > main()
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main(){
7  vector<string> bookNames = {"Book1", "Book2", "Book3", "Book4"};
8  vector<bool> isAvaliable(bookNames.size(), true);
9
10 bool exit = false;
11 int takeChoice = 0;
12 int returnChoice = 0;
13
14 while (!exit){
15
16     int option = 0;
17
18     MainMenu:
19     cout << "Main menu.\n" << "1) list all books\n" << "2) take a book\n" << "3) return a book\n" << "4) exit\n";
20
21
22     cin >> option;
23
24     switch (option){
25
26     case 1:
27
28         for (int i = 0; i < 4; i++){
29
30             cout << i+1 << " ) \"" << bookNames[i] << "\"\n" << " ";
31
32             if (isAvaliable[i]){
33                 cout << "avaliabile\n" << endl;
34             }
35
36             else{
37                 cout << "unavaliabile\n" << endl;
38             }
39         }
40
41         break;
42
43     case 2:
44
45         TakeBook:
46
47         cout << "Enter the number of book you would like to take:\n(from 1 to 4)\n";
48     }
```

```

48
49     cin >> takeChoice;
50
51     if (takeChoice > 0 && takeChoice < 5){
52         if (isAvaliable[takeChoice-1]) {
53             cout << "You have successsfully borrowed the book \"" << bookNames[takeChoice-1] << "\"!\n\n";
54             isAvaliable[takeChoice-1] = 0;
55         }
56
57         else {
58             cout << "You have already borrowed the book \"" << bookNames[takeChoice-1] << "\"!\n\n";
59         }
60     }
61
62     else {
63         cout << "Incorrect input!\n\n";
64
65         goto TakeBook;
66     }
67
68     break;
69
70 case 3:
71
72     ReturnBook:
73
74     cout << "Enter the number of book you would like to return:\n(from 1 to 4)\n";
75
76     cin >> returnChoice;
77
78     if (returnChoice > 0 && returnChoice < 5){
79         if (!isAvaliable[returnChoice-1]) {
80             cout << "You have successsfully returned the book \"" << bookNames[returnChoice-1] << "\"!\n\n";
81             isAvaliable[returnChoice-1] = 1;
82         }
83
84         else {
85             cout << "The library already contains the book \"" << bookNames[returnChoice-1] << "\"!\n\n";
86         }
87     }
88
89     else {
90         cout << "Incorrect input!\n\n";
91

```

```

        goto ReturnBook;
    }

    break;
case 4:

    cout << "exiting the program!\n";

    exit = true;

    break;

default:

    cout << "Incorrect input!\n";

    goto MainMenu;

    break;
}

if (exit) {
    break;
}

char otherOps;

do {

    cout << "Do you want to perform any other operations? (y/n)\n";

    cin >> otherOps;

    if (otherOps == 'n' || otherOps == 'N'){
        exit = true;
    }

    break;

} while (otherOps != 'n' && otherOps != 'N' && otherOps != 'y' && otherOps != 'Y');

}

```

## Завдання №2:

### VNS Lab 2 - Task 1-25

```

1  #include <iostream>
2  #include <cmath>
3
4  #define e 2.71828
5
6  using namespace std;
7
8  int main () {
9
10     double sum = 0.0;
11
12     for (double n = 1.0; n < 8; n++) {
13
14         double br = (-1) * sqrt(n);
15
16         double a_n = (pow(n, 2) * pow(e, br));
17
18         sum += a_n;
19
20         cout << n << " ) " << a_n << endl << sum << endl;
21     }
22
23     return 0;
24 }

```

## Завдання №3:

### VNS Lab 3 - Task 1-25

```

1  #include <iostream>
2  #include <cmath>
3  #include <iomanip>
4
5  using namespace std;
6
7
8  double exactFunction(double x) {
9      return (exp(x) - exp(-x)) / 2;
10 }
11
12
13 double seriesWithFixedNRecursive(double x, int n, int i = 0) {
14
15     if (i > n) {
16         return 0.0;
17     }
18
19     double term = pow(x, 2 * i + 1) / tgamma(2 * i + 2);
20
21     return term + seriesWithFixedNRecursive(x, n, i + 1);
22 }
23
24
25 double seriesWithEpsilon(double x, double epsilon) {
26     double sum = 0.0;
27     double term = x;
28     int i = 0;
29
30     while (fabs(term) > epsilon) {
31         sum += term;
32         i++;
33         term = pow(x, 2 * i + 1) / tgamma(2 * i + 2);
34     }
35     return sum;
36 }
37
38 int main() {
39     double a = 0.1;
40     double b = 1.0;
41     int k = 10;
42     int n = 20;
43     double epsilon = 0.0001;
44
45     double step = (b - a) / k;
46
47     cout << fixed << setprecision(6);
48     cout << " x\tSeries (n = " << n << ")\t\tSeries (Eps = " << epsilon << ")\tExact Value\n";
49
50     cout << fixed << setprecision(6);
51     cout << " x\tSeries (n = " << n << ")\t\tSeries (Eps = " << epsilon << ")\tExact Value\n";
52     cout << "-----\n";
53
54     for (double x = a; x <= b; x += step) {
55         double valueWithN = seriesWithFixedNRecursive(x, n);
56         double valueWithEpsilon = seriesWithEpsilon(x, epsilon);
57         double exactValue = exactFunction(x);
58
59         cout << x << "\t" << valueWithN << "\t\t" << valueWithEpsilon << "\t\t" << exactValue << "\n";
60     }
61
62     return 0;
63 }

```

**Завдання №4:**

**[VNS Lab 7 - Task 1-25](#)**

```

1  #include <iostream>
2  #include <cmath>
3  #include <vector>
4  #include <utility>
5  #include <cstdlib>
6
7  using namespace std;
8
9
10 double distance(double x1, double y1, double x2, double y2) {
11     return sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1));
12 }
13
14 double square(double x1, double y1, double x2, double y2, double x3, double y3) {
15     double a = distance(x1, y1, x2, y2);
16     double b = distance(x1, y1, x3, y3);
17     double c = distance(x3, y3, x2, y2);
18
19     double p = (a + b + c)/2.0;
20
21     return sqrt(p*(p - a)*(p - b)*(p - c));
22 }
23
24 double square1(int num_cords, ...) {
25     if (num_cords < 3) {
26         cerr << "Not enough points to form a polygon" << endl;
27         return 0;
28     }
29
30     va_list args;
31     va_start(args, num_cords);
32
33     vector<int> cords;
34     for (int i = 0; i < num_cords; i++) {
35         int x = va_arg(args, int);
36         cords.push_back(x);
37     }
38     va_end(args);
39
40     vector<pair<double, double>> points;
41     for (int i = 0; i < num_cords/2; i++) {
42         double x = cords[i];
43         double y = cords[i + 1];
44         points.emplace_back(x, y);
45     }
46
47     int n = points.size();
48     double max_diagonal = 0;
49     pair<int, int> max_diagonal_points = {0, 1};
50
51     for (int i = 0; i < n; i++) {
52         for (int j = i + 1; j < n; j++) {
53             double d = distance(points[i].first, points[i].second, points[j].first, points[j].second);
54             if (d > max_diagonal) {
55                 max_diagonal = d;
56                 max_diagonal_points = {i, j};
57             }
58         }
59     }
60
61     double max_area = 0;
62     int a = max_diagonal_points.first;
63     int b = max_diagonal_points.second;
64
65     for (int i = 0; i < n; i++) {
66         if (i != a && i != b) {
67             double area = square(points[a].first, points[a].second,
68                                 points[b].first, points[b].second,
69                                 points[i].first, points[i].second);
70             if (area > max_area) {
71                 max_area = area;
72             }
73         }
74     }
75
76     return max_area;
77 }
78
79 int main() {
80
81     vector<pair<double, double>> points = {{0, 0}, {5, 0}, {3, 4}, {6, 6}};
82
83     cout << "Edge length: " << distance(0.0, 0.0, 5.0, 0.0) << endl;
84     cout << "Area of a triangle: " << square(0, 0, 5, 0, 3, 4) << endl;
85     cout << "Area of the triangle with the longest diagonal: " << square1(8, 0, 0, 5, 0, 3, 4, 6, 6) << endl;
86
87     return 0;
88 }
89

```

**Завдання №5:**

**VNS Lab 7 - Task 2-25**

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6
7  void deleteEvenRows(vector<vector<int>>& matrix) {
8      for (int i = 0; i < matrix.size(); ++i) {
9          if (i % 2 == 1) {
10             matrix[i].clear();
11         }
12     }
13 }
14
15
16 void deleteBetweenZeros(vector<int>& arr) {
17     int firstZero = -1, secondZero = -1;
18
19     for (int i = 0; i < arr.size(); ++i) {
20         if (arr[i] == 0) {
21             if (firstZero == -1) {
22                 firstZero = i;
23             } else {
24                 secondZero = i;
25             }
26         }
27     }
28
29     if (firstZero != -1 && secondZero != -1) {
30         arr.erase(arr.begin() + firstZero + 1, arr.begin() + secondZero);
31     }
32
33 }
34
35
36
37 }
38
39 void print (vector<vector<int>> vec2d) {
40     for (const auto& row : vec2d) {
41         if (!row.empty()) {
42             for (int val : row) {
43                 cout << val << " ";
44             }
45             cout << endl;
46         }
47     }
48 }

```



```

48 }
49
50 void print (vector<int> vec1d) {
51     for (int val : vec1d) {
52         cout << val << " ";
53     }
54 }
55
56 int main() {
57
58     vector<vector<int>> matrix = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}};
59
60     cout << "Original 2-dimensional arr: " << endl;
61     print(matrix);
62
63     deleteEvenRows(matrix);
64
65     cout << "\nWith lines deleted:" << endl;
66     print(matrix);
67
68     vector<int> arr = {1, 2, 0, 3, 4, 0, 5, 6};
69
70     cout << "\nOriginal arr: ";
71     print(arr);
72     cout << endl;
73
74     deleteBetweenZeros(arr);
75
76     cout << "After deleting elements between two 0's: ";
77     print(arr);
78     cout << endl;
79
80     return 0;
81 }
82

```

**Результат виконання завдань, тестування та фактично  
затрачений час:**

**Завдання №1:**

**Class Practice Task**

Фактично затрачений час: 2,5 години.

```
Main menu.  
1) list all books  
2) take a book  
3) return a book  
4) exit  
5  
Incorrect input!  
Main menu.  
1) list all books  
2) take a book  
3) return a book  
4) exit  
1  
1) "Book1" available  
  
2) "Book2" available  
  
3) "Book3" available  
  
4) "Book4" available  
  
Do you want to perform any other operations? (y/n)  
y  
Main menu.  
1) list all books  
2) take a book  
3) return a book  
4) exit  
2  
Enter the number of book you would like to take:  
(from 1 to 4)  
2  
You have successfully borrowed the book "Book2"!  
  
Do you want to perform any other operations? (y/n)  
y  
Main menu.  
1) list all books  
2) take a book  
3) return a book  
4) exit  
3  
Enter the number of book you would like to return:  
(from 1 to 4)  
3  
The library already contains the book "Book3"!  
  
Do you want to perform any other operations? (y/n)  
y  
Main menu.  
1) list all books  
2) take a book  
3) return a book  
4) exit  
4
```

## Завдання №2:

### VNS Lab 2 - Task 1-1

Фактично затрачений час: 5 хвилин.

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS
1)	0.36788			
	0.36788			
2)	0.972468			
	1.34035			
3)	1.59229			
	2.93264			
4)	2.16537			
	5.09801			
5)	2.67195			
	7.76996			
6)	3.10816			
	10.8781			
7)	3.47666			
	14.3548			

## Завдання №3:

### VNS Lab 3 - Task 1-1

Фактично затрачений час: 45 хвилин.

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS
x	Series (n = 20)	Series (Eps = 0.000100)	Exact Value	
0.100000	0.100167	0.100167	0.100167	
0.190000	0.191145	0.191143	0.191145	
0.280000	0.283673	0.283659	0.283673	
0.370000	0.378500	0.378442	0.378500	
0.460000	0.476395	0.476394	0.476395	
0.550000	0.578152	0.578149	0.578152	
0.640000	0.684594	0.684585	0.684594	
0.730000	0.796586	0.796564	0.796586	
0.820000	0.915034	0.914984	0.915034	
0.910000	1.040899	1.040898	1.040899	
1.000000	1.175201	1.175198	1.175201	

## Завдання №4:

### VNS Lab 7 - Task 1-1

Фактично затрачений час: година.

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS
			Edge length: 5	
			Area of a triangle: 10	
			Area of the triangle with the longest diagonal: 15	
			PS C:\VS\projects>	

## Завдання №5:

### VNS Lab 7 - Task 2-1

Фактично затрачений час: година.

```
34
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
```

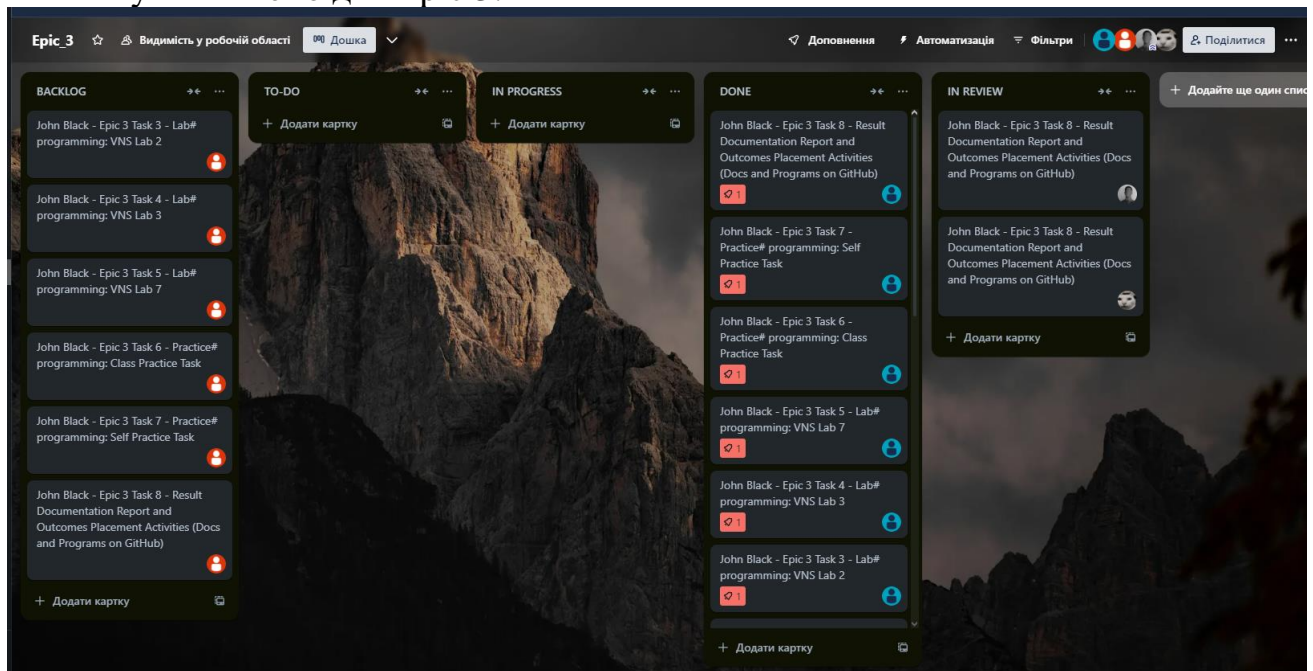
```
Original 2-dimensional arr:
1 2 3
4 5 6
7 8 9
10 11 12

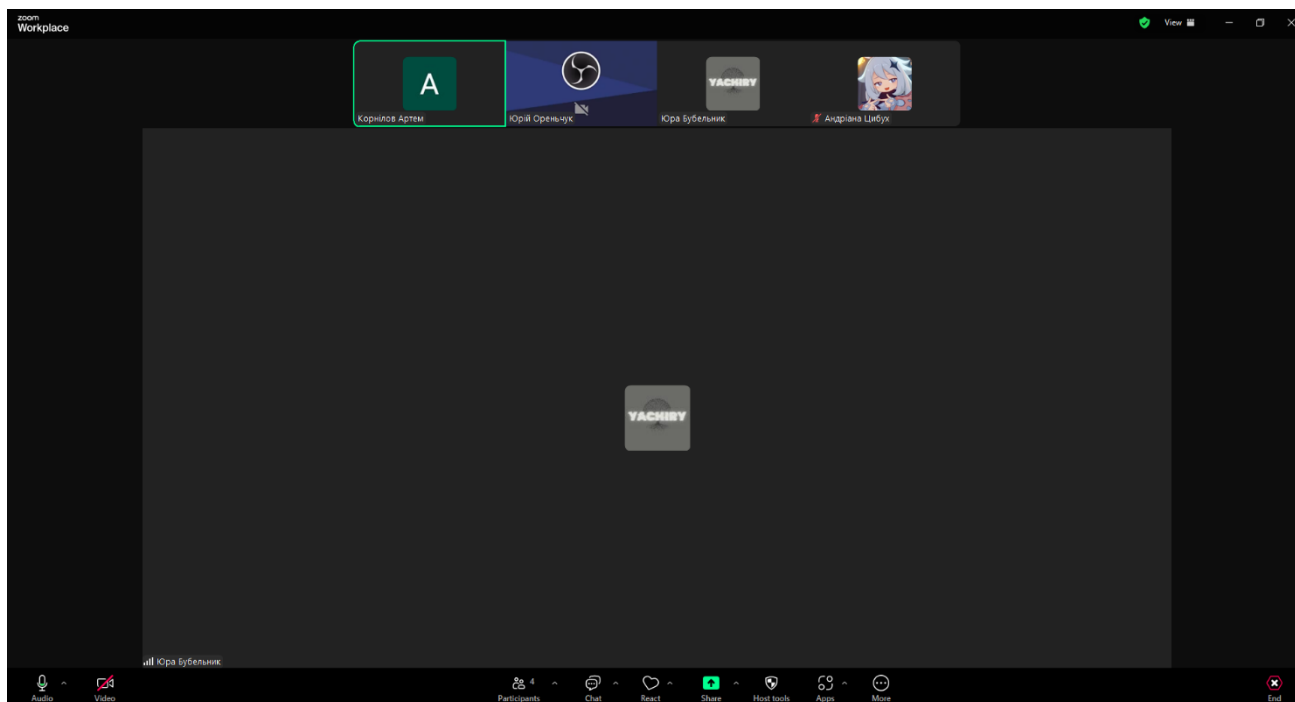
With lines deleted:
1 2 3
7 8 9

Original arr: 1 2 0 3 4 0 5 6
After deleting elements between two 0's: 1 2 0 0 5 6
```

## Робота з командою:

Налаштували Trello для Epic 3:





## Висновки:

Отже, я навчився використовувати перезавантажені функції та покращив роботу з циклами й їх застосуваннями. Також я ознайомився з реалізацією комплексних чисел в програмуванні та з базовими математичними функціями.

[Посилання на pull request](#)