

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-12

Горішний Микола Мар'янович

Львів 2024

Тема роботи:

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

Мета роботи:

Навчитися писати основні функції, цикли в C++. Розібрати що таке простори імен та як їх використовувати. Що таке перевантажені функції та навіщо вони потрібні, навчитись будувати блок-схеми складнішого типу.

Джерела інформації:

<https://acode.com.ua/urok-66-operator-y-upravlinnya-potokom-vykonannya-program/>;

<https://uk.javascript.info/while-for>;

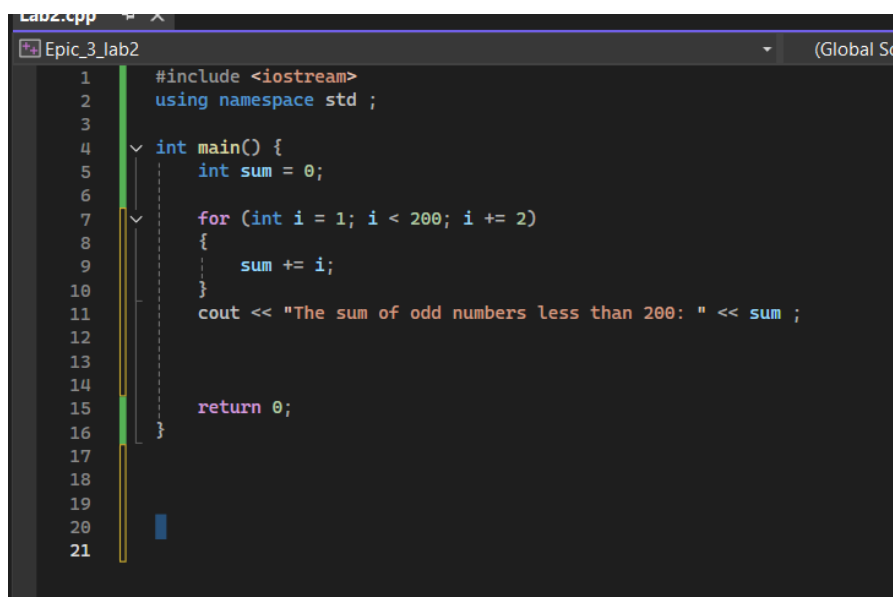
<https://www.youtube.com/watch?v=Rd10XYY94dI>;

https://www.youtube.com/watch?v=V7q9w_s0nns;

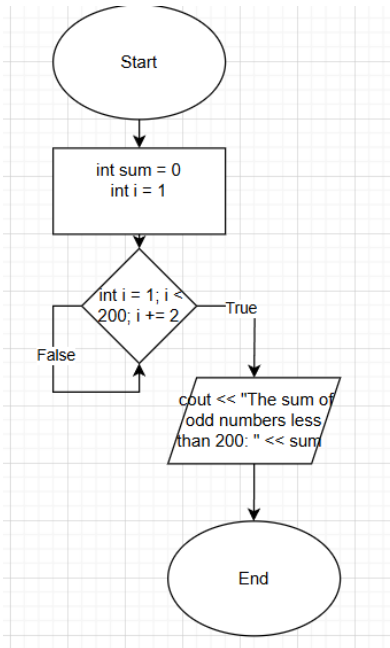
Виконання роботи:

Lab# programming: VNS Lab 2 Task 1:

3) Знайти суму цілих додатніх непарних чисел, менших 200.



```
1 #include <iostream>
2 using namespace std ;
3
4 int main() {
5     int sum = 0;
6
7     for (int i = 1; i < 200; i += 2)
8     {
9         sum += i;
10    }
11    cout << "The sum of odd numbers less than 200: " << sum ;
12
13
14
15    return 0;
16 }
```



Lab# programming: 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:

- для заданого n ;
- для заданої точності ϵ ($\epsilon=0.0001$).

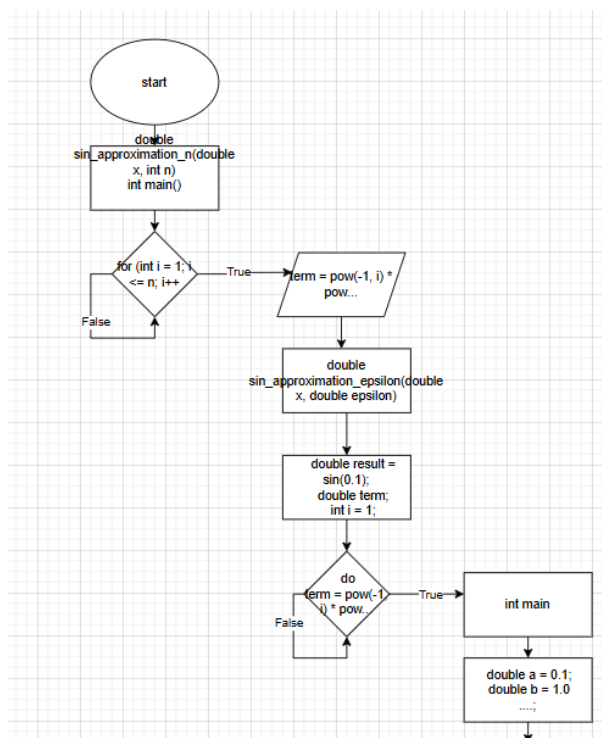
Для порівняння знайти точне значення функції.

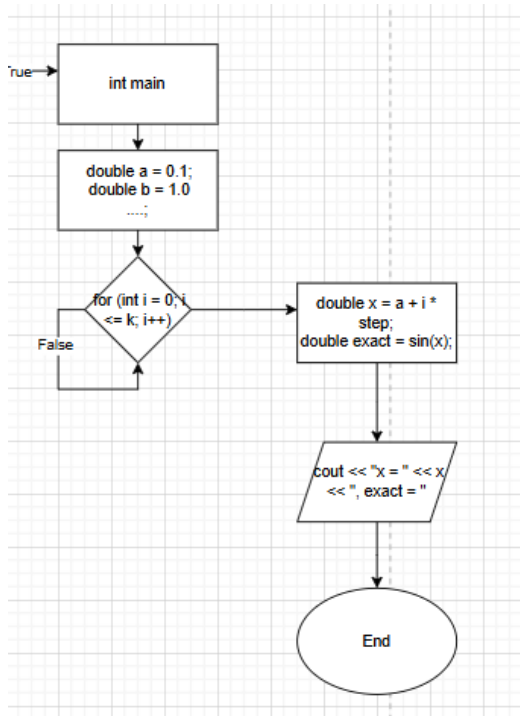
3	$y = \sin X$	$0,1 \leq x \leq 1$	10	$S = x - \frac{x^3}{3!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$
---	--------------	---------------------	----	--

```

1 #include <iostream>
2 #include <cmath>
3 #include <iomanip>
4 using namespace std;
5
6 double sin_approximation_n(double x, int n)
7 {
8     double result = sin(0.1);
9     double term;
10
11     for (int i = 1; i <= n; i++) {
12         term = pow(-1, i) * pow(x - 0.1, 2 * i + 1) / tgamma(2 * i + 2); // (2*i+1)! = tgamma(2*i+2)
13         result += term;
14     }
15
16     return result;
17 }
18
19 double sin_approximation_epsilon(double x, double epsilon)
20 {
21     double result = sin(0.1);
22     double term;
23     int i = 1;
24
25     do {
26         term = pow(-1, i) * pow(x - 0.1, 2 * i + 1) / tgamma(2 * i + 2); // tgamma = факторіал
27         result += term;
28         i++;
29     } while (fabs(term) >= epsilon); // робимо поки величина не буде менше ніж epsilon
30
31     return result;
32 }
33
34 int main() {
35     double a = 0.1;
36     double b = 1.0;
37     int k = 10; // крок
38     int n = 5; // Кількість йифрів для апрокскації
39     double epsilon = 0.0001; // наша точність
40
41     int n = 5; // Кількість йифрів для апрокскації
42     double epsilon = 0.0001; // наша точність
43
44     double step = (b - a) / k;
45
46     cout << fixed << setprecision(6);
47
48     for (int i = 0; i <= k; i++) {
49         double x = a + i * step;
50
51         double exact = sin(x);
52
53         double approx_n = sin_approximation_n(x, n);
54         double approx_epsilon = sin_approximation_epsilon(x, epsilon);
55
56         cout << "x = " << x << ", exact = " << exact << ", approx (n=" << n << ") = " << approx_n
57             << ", approx (ε=" << epsilon << ") = " << approx_epsilon << endl;
58     }
59
60     return 0;
61 }

```





Lab# programming: VNS Lab 7 Task 2

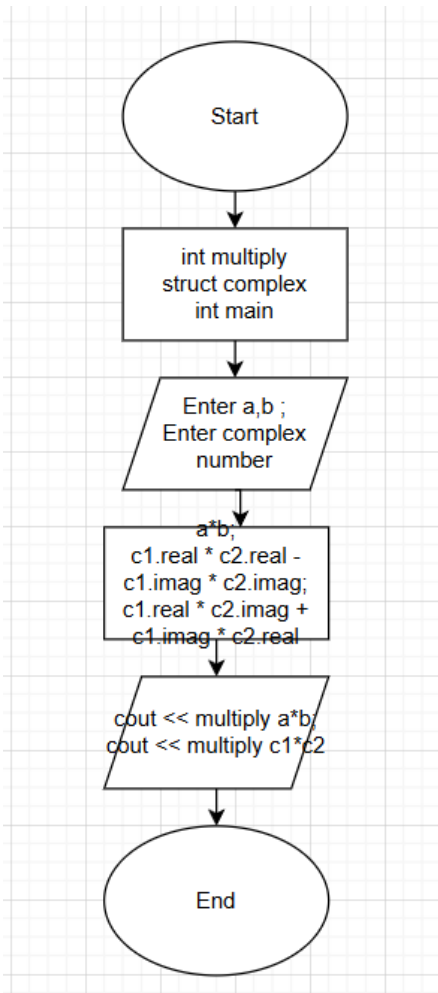
3.

- а) для множення цілих чисел;
- б) для множення комплексних чисел.

```

Epic_3_Multiply (Global Scope)
1  #include <iostream>
2  using namespace std;
3
4
5  int multiply(int a, int b)
6  {
7      return a * b;
8  }
9
10
11 struct Complex
12 {
13     int real;
14     int imag;
15 };
16
17 Complex multiply(Complex c1, Complex c2) {
18     Complex result;
19     result.real = c1.real * c2.real - c1.imag * c2.imag;
20     result.imag = c1.real * c2.imag + c1.imag * c2.real;
21     return result;
22 }
23
24
25 void printComplex(Complex c) {
26     cout << c.real << " + " << c.imag << "i" << endl;
27 }
28
29 int main() {
30
31     int a, b;
32     cout << "Enter first number :";
33     cin >> a;
34     cout << "Enter second number: ";
35     cin >> b;
36     cout << "multiplication of full number : " << multiply(a, b) << endl;
37
38     // cout << "multiplication of full number : " << multiply(a, b) << endl;
39
40     Complex c1, c2;
41     cout << "Enter full and imaginary part of complex number for second number : ";
42     cin >> c1.real >> c1.imag;
43     cout << "Enter full and imaginary part of complex number for first number : ";
44     cin >> c2.real >> c2.imag;
45
46
47     Complex result = multiply(c1, c2);
48     cout << "multiplication of complex number: ";
49     printComplex(result);
50
51
52
53     return 0;
54 }
55

```

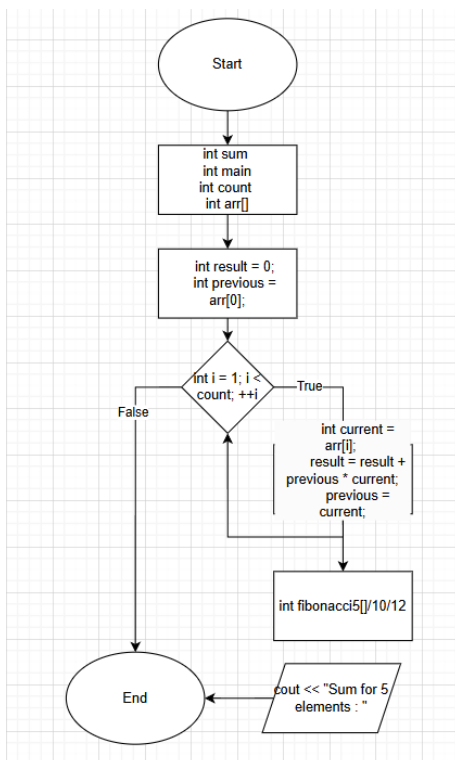


Lab# programming: VNS Lab 7 Task 1

3. Написати функцію sum зі змінною кількістю параметрів, що знаходить суму чисел типу int за формулою:

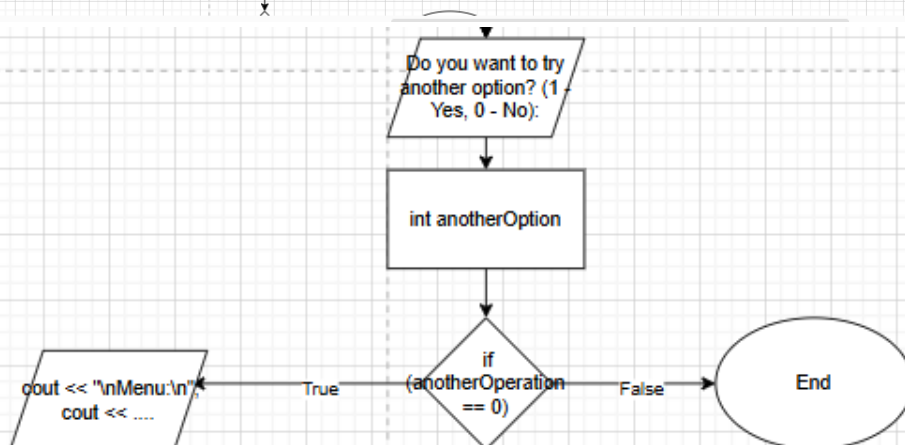
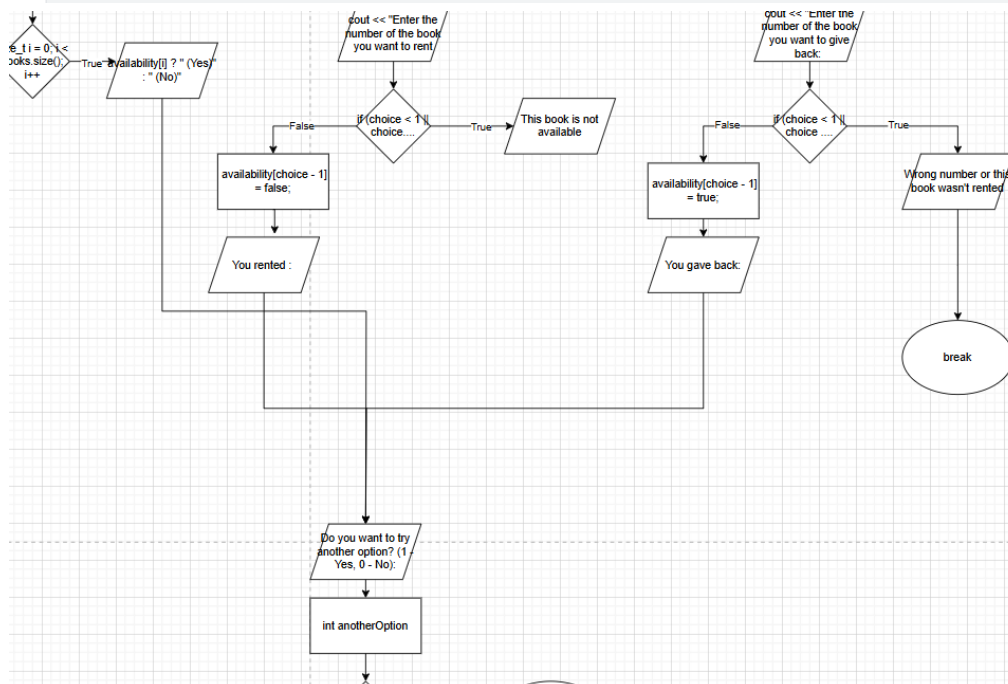
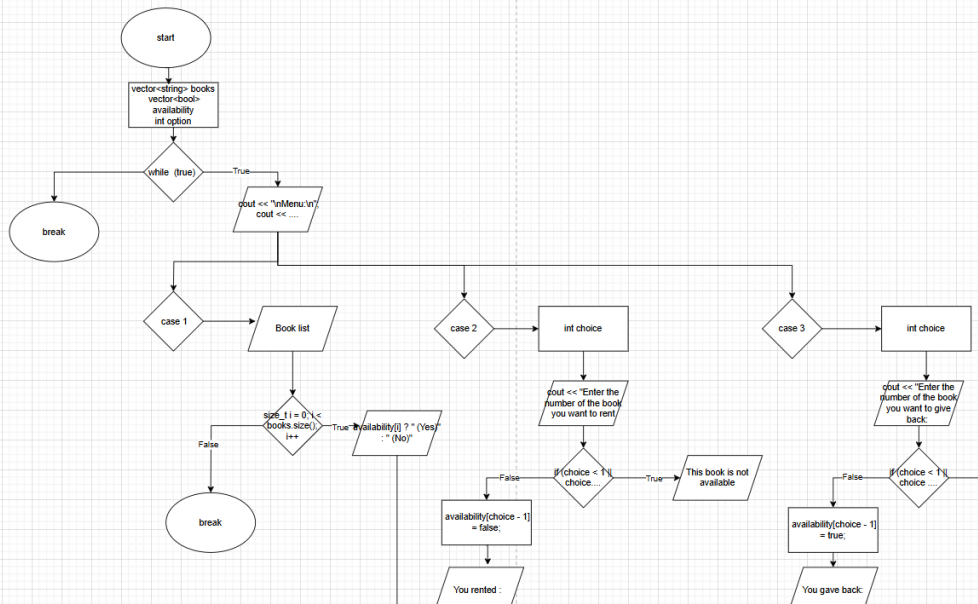
$$S = a_1 * a_2 + a_2 * a_3 + a_3 * a_4 + \dots$$

```
Epic_3_Task1 Fibonacci (Global Scope)
1  #include <iostream>
2  using namespace std;
3
4
5  int sum(int count, int arr[])
6  {
7
8      int result = 0;
9      int previous = arr[0];
10
11     for (int i = 1; i < count; ++i) {
12         int current = arr[i];
13         result = result + previous * current;
14         previous = current;
15     }
16
17     return result;
18 }
19
20 int main()
21 {
22     int fibonacci5[] = { 1, 1, 2, 3, 5 };
23     int fibonacci10[] = { 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 };
24     int fibonacci12[] = { 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 };
25
26
27     cout << "Sum for 5 elements Fibonacci: " << sum(5, fibonacci5) << endl;
28     cout << "Sum for 10 elements Fibonacci : " << sum(10, fibonacci10) << endl;
29     cout << "Sum for 12 elements Fibonacci : " << sum(12, fibonacci12) << endl;
30
31
32
33
34
35
36     return 0;
37 }
38
39
```



Practice# programming: Class Practice Task

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4
5 using namespace std;
6
7 int main() {
8     vector<string> books = { "First", "Second", "Third", "Fourth", "Fifth" };
9     vector<bool> availability = { true, true, true, true, true };
10
11     int option;
12
13     while (true) {
14         cout << "\nMenu:\n";
15         cout << " 1 - Count all books \n";
16         cout << " 2 - Rent a book \n";
17         cout << " 3 - Give back a book \n";
18         cout << " 4 - Exit \n";
19         cout << "Choose your option: ";
20         cin >> option;
21
22         switch (option) {
23             case 1: {
24                 cout << "Book list: \n";
25                 for (size_t i = 0; i < books.size(); i++) {
26                     cout << (i + 1) << ". " << books[i] << (availability[i] ? " (Yes)" : " (No)") << endl;
27                 }
28                 break;
29             }
30             case 2: {
31                 int choice;
32                 cout << "Enter the number of the book you want to rent: ";
33                 cin >> choice;
34                 if (choice < 1 || choice > books.size() || !availability[choice - 1]) {
35                     cout << "This book is not available.\n";
36                     break;
37                 }
38
39                 availability[choice - 1] = false;
40                 cout << "You rented: " << books[choice - 1] << "\n";
41                 break;
42             }
43             case 3: {
44                 int choice;
45                 cout << "Enter the number of the book you want to give back: ";
46                 cin >> choice;
47                 if (choice < 1 || choice > books.size() || availability[choice - 1]) {
48                     cout << "Wrong number or this book wasn't rented.\n";
49                     break;
50                 }
51
52                 availability[choice - 1] = true;
53                 cout << "You gave back: " << books[choice - 1] << "\n";
54                 break;
55             }
56             case 4: {
57                 cout << "Exit complete \n";
58                 return 0;
59             }
60             default: {
61                 cout << "Invalid option, please try again.\n";
62             }
63
64             cout << "Do you want to try another option? (1 - Yes, 0 - No): ";
65             int anotherOperation;
66             cin >> anotherOperation;
67
68             if (anotherOperation == 0) {
69                 break;
70             }
71         }
72
73         return 0;
74     }
75 }
```

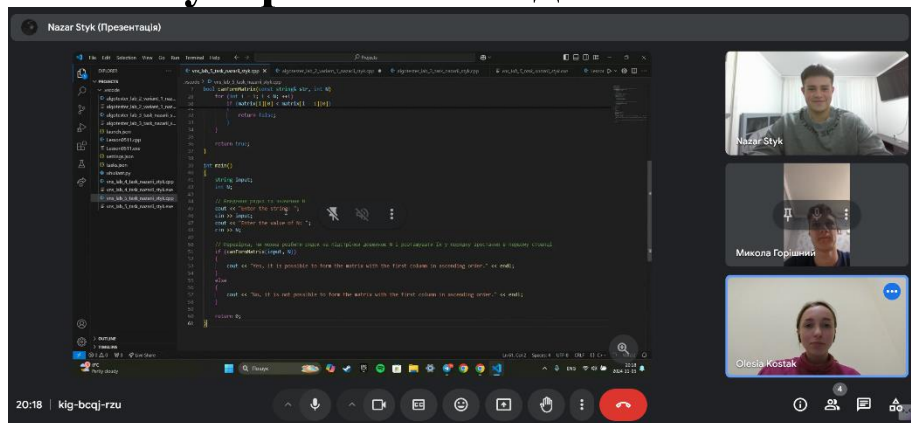


Practice# programming: Self Practice Algotester Task

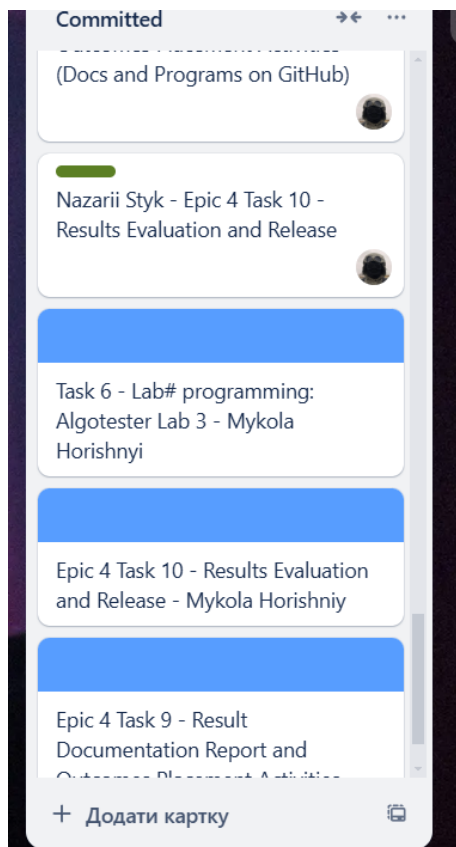
```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a, b, c;
7      cin >> a >> b >> c;
8
9      if (a + b > c) {
10         cout << "YES" << endl;
11     } else {
12         cout << "NO" << endl;
13     }
14
15
16
17     return 0;
18 }
19
```

Result	Time (sec.)	Memory (MiB)	Actions
Accepted	0.002	1.070	View

Наша зустріч з командою:



Trello:



Висновок: я навчився робити складні блок-схеми в draw.io, також використовувати різні типи циклів і керувати ними, навчився використовувати перевантажені функції і простори імен.