

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 3

На тему: «Програмування: алгоритм, програма, код. Системи числення. Двійкова система числення. Розробка та середовище розробки програми.»

з дисципліни: «Основи програмування»

до:

Практичних Робіт до блоку № 3

Виконала:

Студентка групи ШІ-13
Бобринок Ангеліна Вадимівна

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. Введення в Цикли та їх Види в C++:

- Значення та роль циклів у програмуванні.
- Огляд видів циклів: for, while, do-while.
- Синтаксис та основи використання кожного типу циклу.
- Приклади базових циклів для різних задач.

2. Управління Виконанням Циклів:

- Застосування операторів break та continue.
- Умови завершення циклів.
- Передчасне завершення виконання циклу.
- Приклади та вправи з управлінням циклами.

3. Вкладені Цикли:

- Поняття та важливість вкладених циклів.
- Реалізація вкладених циклів: приклади для різних сценаріїв.
- Практичні завдання на вкладені цикли.

4. Основи Функцій у C++:

- Визначення та оголошення функцій.
- Параметри функцій: передача за значенням і за посиланням.
- Параметри за замовчуванням.
- Повернення значень з функцій.
- Приклади створення та використання функцій.

5. Перевантаження Функцій та Простір Імен:

- Концепція перевантаження функцій.
- Правила та приклади перевантаження функцій.
- Поняття та використання просторів імен.
- Вкладені простори імен (C++ 17)
- Роль просторів імен у організації коду.

6. Розширені Можливості Функцій:

- Функції зі змінною кількістю параметрів (еліпсис): синтаксис та приклади.
- Область видимості функції – static, extern.
- Рекурсія: основи, приклади рекурсивних функцій та їх аналіз.
- Передача масивів та об'єктів як параметрів.
- Повернення масивів та об'єктів з функцій.

7. Вбудовані Функції в C++:

- Огляд вбудованих функцій у C++.
- Приклади використання стандартних функцій у програмуванні.
- Роль вбудованих функцій у спрощенні коду.
- Практичні завдання для розуміння вбудованих функцій.

Виконання роботи:

1) *Опрацювання завдання та вимог до програм та середовища:*

Завдання №1

Знайти суму цілих додатніх непарних чисел, менших 200.

Завдання №2

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.

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

Завдання №3

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

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

Написати викликаючу функцію `main`, що звертається до функції `sum` не менше трьох разів з кількістю параметрів 5, 10, 12.

Завдання №4

Написати перевантажені функції й основну програму, що їх викликає.

а) для множення цілих чисел;

б) для множення комплексних чисел.

Завдання №5

Ви створюєте просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

Завдання №6

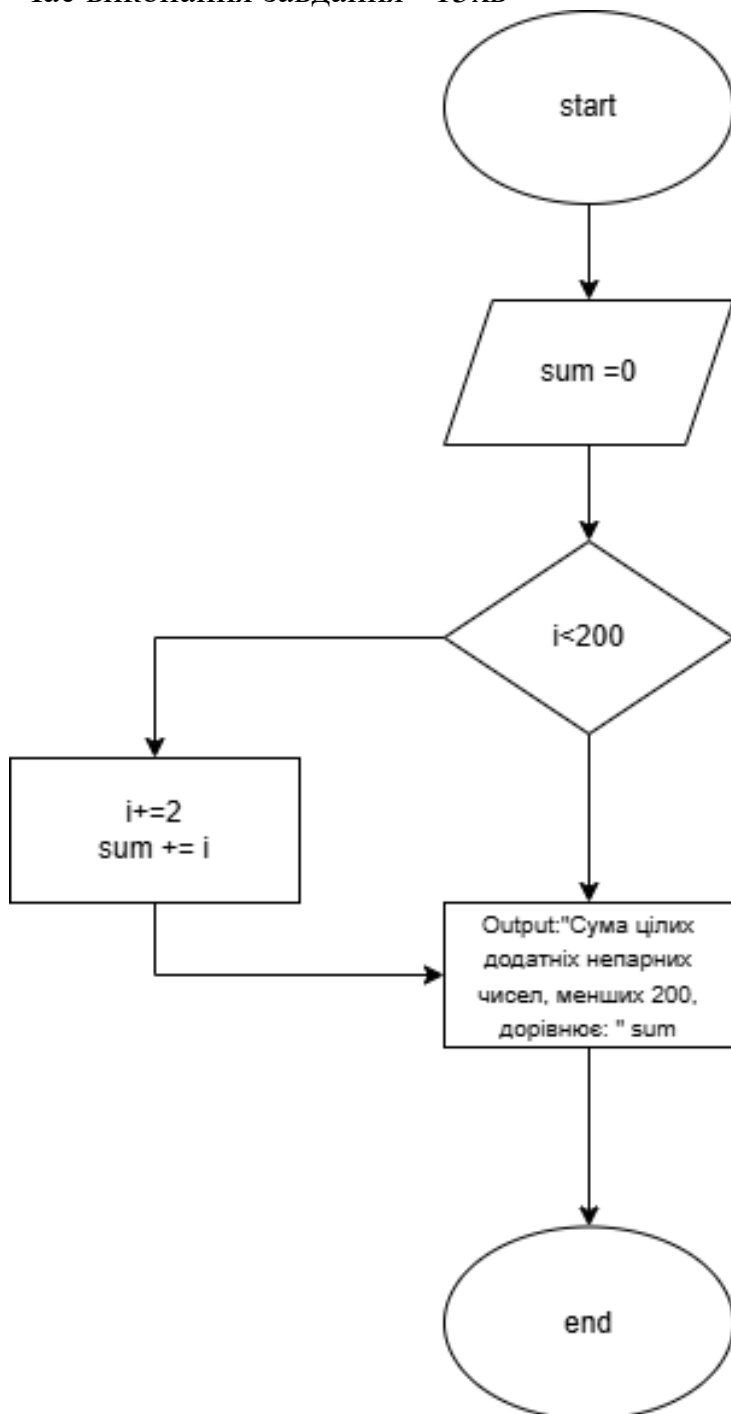
Одного разу до Ужгорода на літню школу з алгоритмічного програмування приїхали n студентів, що сформували k команд. Відомо, що кожна команда складається з одного, двох або трьох студентів.

Вам необхідно визначити, скільки студентів було в кожній із команд.

2) Дизайн та планована оцінка часу виконання завдань:

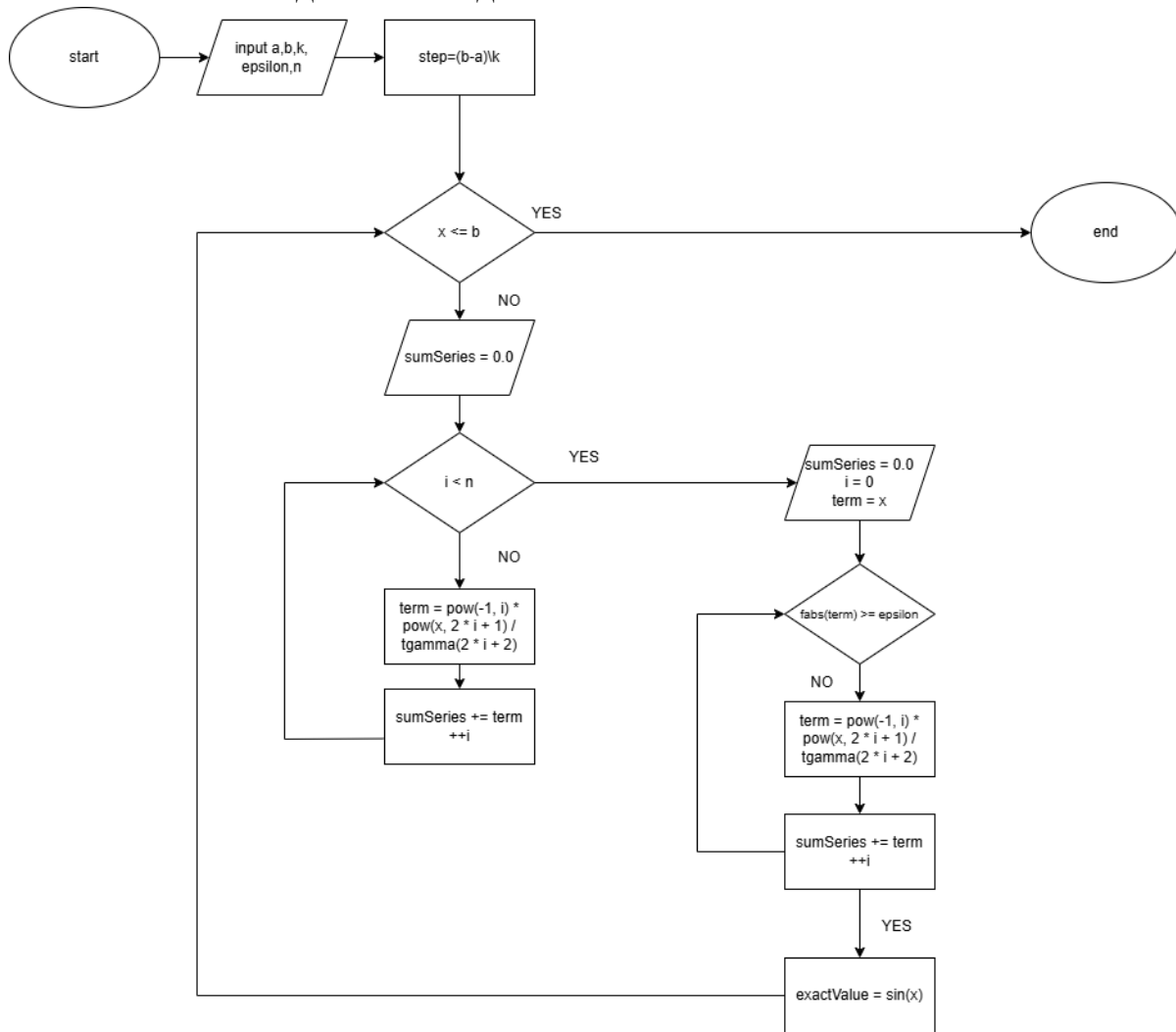
Завдання №1

Час виконання завдання ~15хв



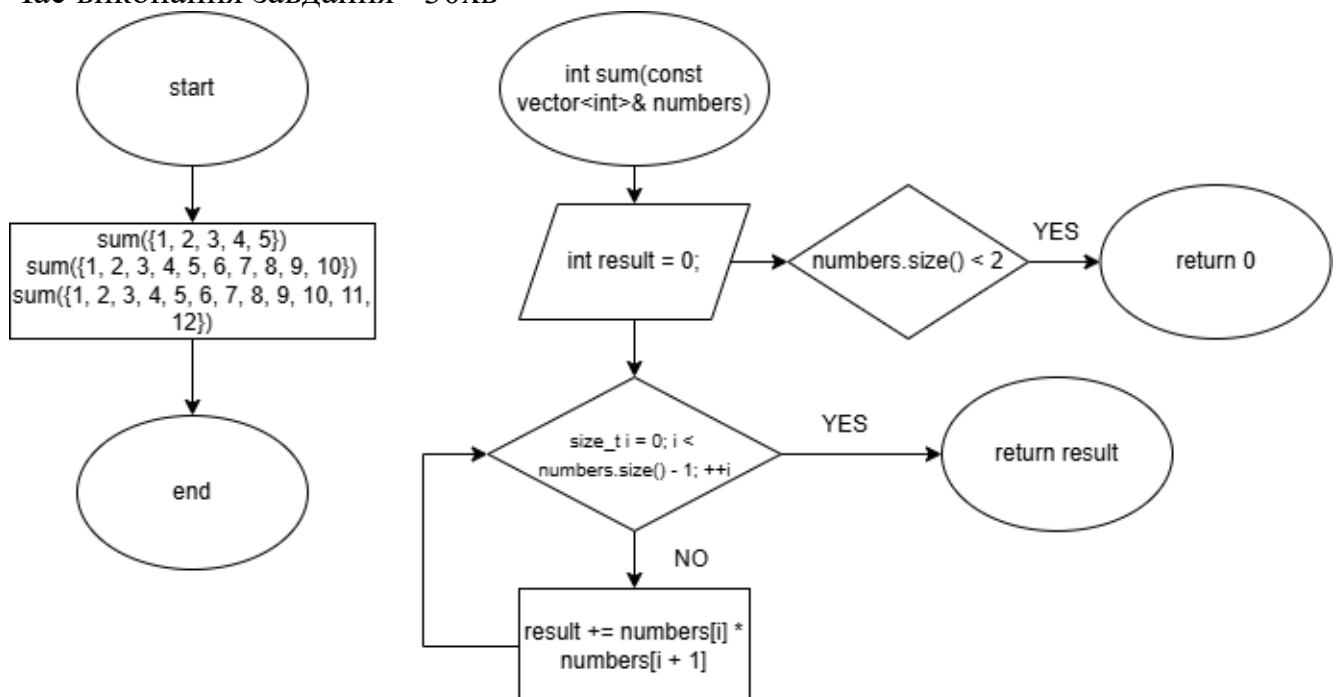
Завдання №2

Час виконання завдання ~1 год



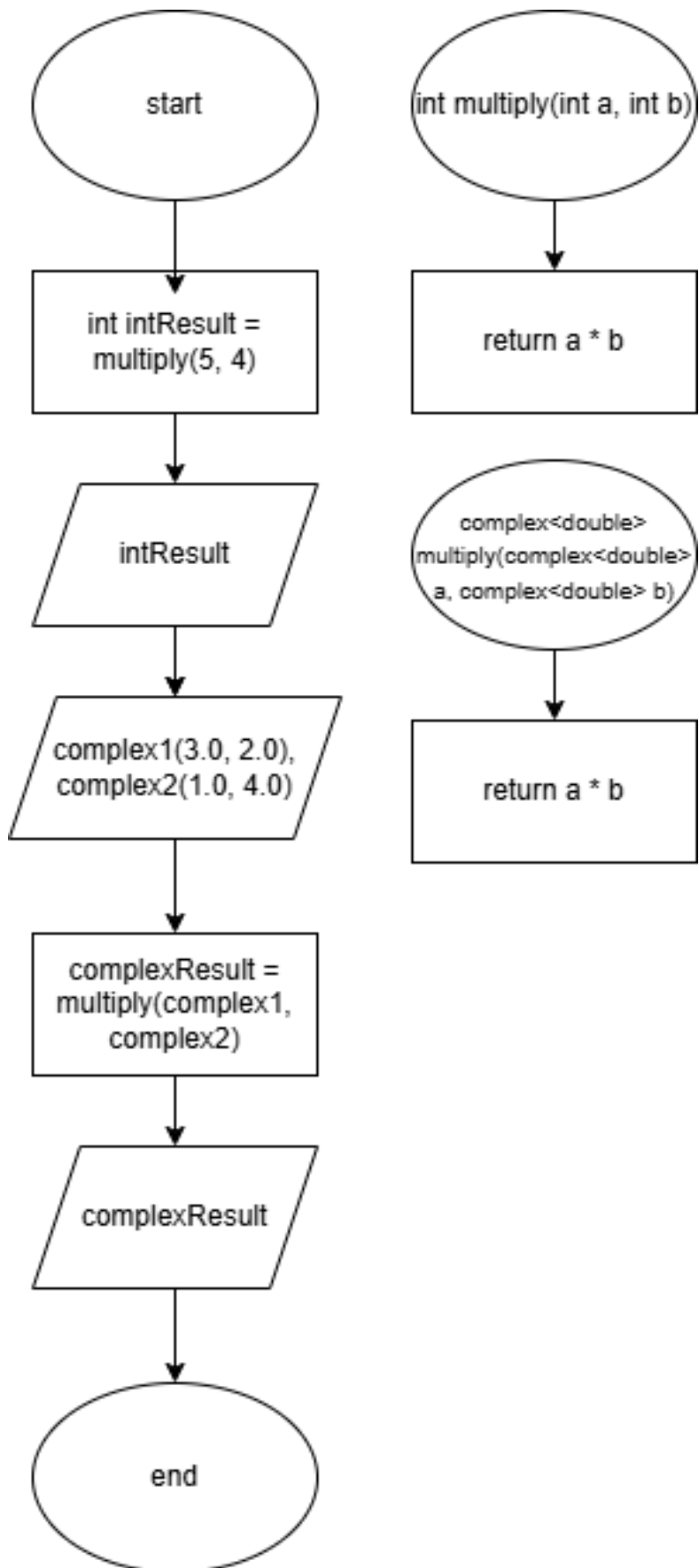
Завдання №3

Час виконання завдання ~30хв



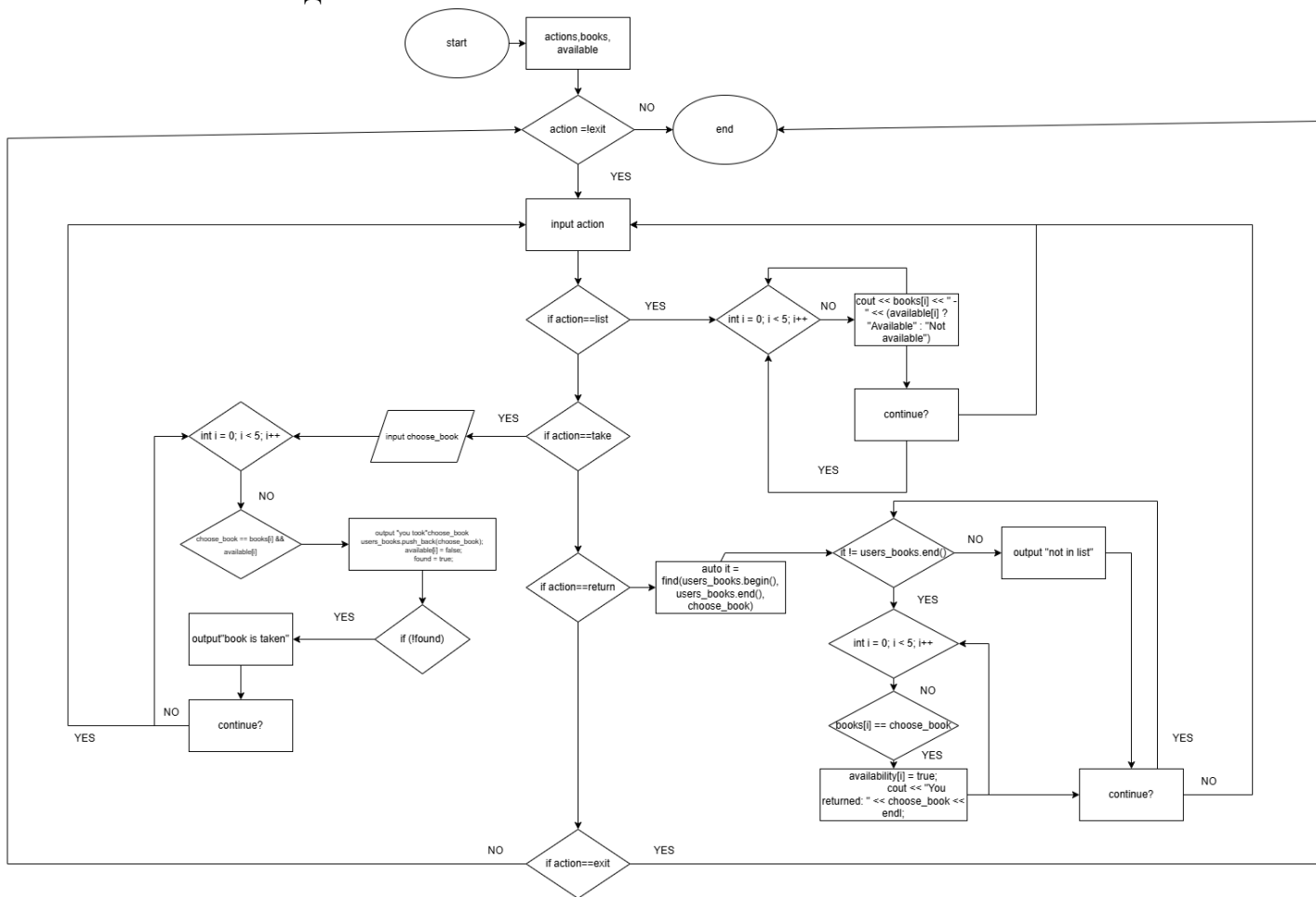
Завдання №4

Час виконання завдання ~20хв



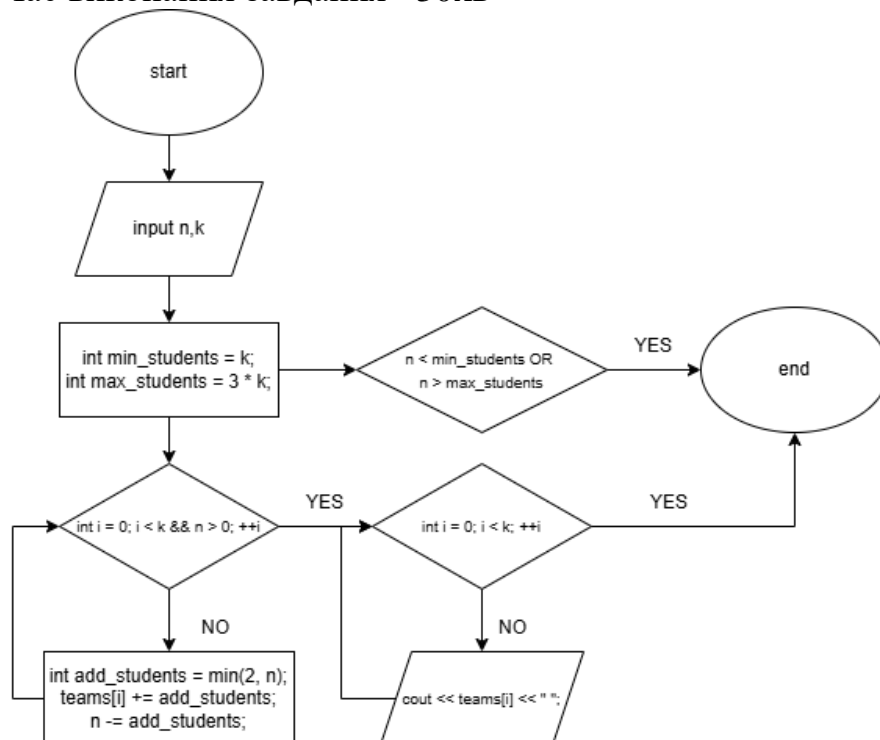
Завдання №5

Час виконання ~1 година



Завдання №6

Час виконання завдання ~30хв

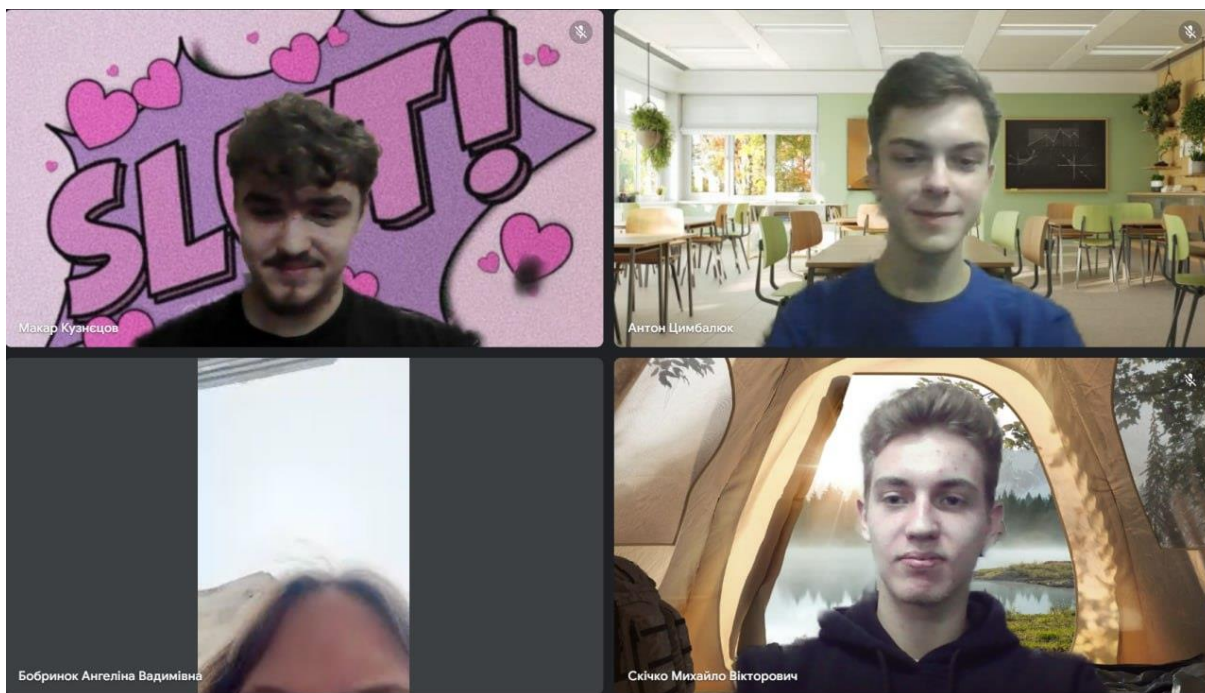


3) Конфігурація середовища до виконання завдань:

Планування роботи в notion

	Task	⚙ Status	📅 deadline
Anhelina	Theory Education Activities	In progress	15/11/2024
Anhelina	Requirements management (understand tasks) and design activities	Done	15/11/2024
Anhelina	Lab3 programming: VNS Lab 2	In progress	15/11/2024
Anhelina	Lab3 programming: VNS Lab 3	Pending	15/11/2024
Anhelina	Lab3 programming: VNS Lab 7	Pending	15/11/2024
Anhelina	Practice3 programming: Class Practice Task	Pending	15/11/2024
Anhelina	Practice3 programming: Self Practice Task	Pending	15/11/2024
Anhelina	Result Documentation Report and Outcomes Placement Activities	Pending	15/11/2024
Anhelina	Results Evaluation and Release	Pending	15/11/2024

Зустріч з командою та обговорення питань



4) Код програм з посиланням на зовнішні ресурси:

Завдання №1

vns_lab_2_anhelina_bobrynok.cpp

```
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          sum += i;
9      }
10
11     cout << "Сума цілих додатніх непарних чисел, менших 200, дорівнює: " << sum << endl;
12     return 0;
13 }
```

Завдання №2

vns_lab_3_anhelina_bobrynok .cpp

```
1  #include <iostream>
2  #include <cmath>
3  #include <iomanip>
4
5  using namespace std;
6
7  // Функція для обчислення ряду при заданій кількості членів n
8  double calculateSeriesFixedN(double x, int n) {
9      double sumSeries = 0.0;
10     for (int i = 0; i < n; ++i) {
11         double term = pow(-1, i) * pow(x, 2 * i + 1) / tgamma(2 * i + 2); //
12         sumSeries += term;
13     }
14     return sumSeries;
15 }
16
17 // Функція для обчислення ряду з точністю epsilon
18 double calculateSeriesPrecision(double x, double epsilon) {
19     double sumSeries = 0.0;
20     int i = 0;
21     double term = x;
22     while (fabs(term) >= epsilon) {
23         term = pow(-1, i) * pow(x, 2 * i + 1) / tgamma(2 * i + 2); //  $(-1)^i * x^{(2i+1)} / (2i+1)!$ 
24         sumSeries += term;
25         ++i;
26     }
27     return sumSeries;
28 }
29
30 int main() {
31
32     double a = 0.1;
33     double b = 1.0;
34     int k = 10;
35     double epsilon = 0.0001;
36     int n = 10;
37
38     double step = (b - a) / k;
39
40     cout << fixed << setprecision(6);
41     cout << " x\t S(x) для n S(x) для ε Точне значення\n";
42     cout << "-----\n";
43
44     for (double x = a; x <= b; x += step) {
45
46         double seriesFixedN = calculateSeriesFixedN(x, n);
47
48         double seriesPrecision = calculateSeriesPrecision(x, epsilon);
49
50         double exactValue = sin(x);
51
52         cout << x << "\t" << seriesFixedN << "\t" << seriesPrecision << "\t" << exactValue << "\n";
53     }
54
55     return 0;
56 }
```

Завдання №3

vns_lab_7_task_1_anhelina_bobrynok .cpp

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int sum(const vector<int>& numbers) {
7      int result = 0;
8
9      if (numbers.size() < 2) {
10         return 0;
11     }
12
13     for (size_t i = 0; i < numbers.size() - 1; ++i) {
14         result += numbers[i] * numbers[i + 1];
15     }
16     return result;
17 }
18
19 int main() {
20     cout << "Sum for 5 parameters: " << sum({1, 2, 3, 4, 5}) << endl;
21     cout << "Sum for 10 parameters: " << sum({1, 2, 3, 4, 5, 6, 7, 8, 9, 10}) << endl;
22     cout << "Sum for 12 parameters: " << sum({1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}) << endl;
23
24     return 0;
25 }
```

Завдання №4

vns_lab_1_task_2_anhelina_bobrynok .cpp

```
1  #include <iostream>
2  #include <complex>
3
4  using namespace std;
5
6  int multiply(int a, int b) {
7      return a * b;
8  }
9
10 complex<double> multiply(complex<double> a, complex<double> b) {
11     return a * b;
12 }
13
14 int main() {
15     int intResult = multiply(5, 4);
16     cout << "Множення цілих чисел 5 * 4 = " << intResult << endl;
17
18     complex<double> complex1(3.0, 2.0); // Комплексне число 3 + 2i
19     complex<double> complex2(1.0, 4.0); // Комплексне число 1 + 4i
20
21     complex<double> complexResult = multiply(complex1, complex2);
22     cout << "Множення комплексних чисел (3 + 2i) * (1 + 4i) = "
23         << complexResult << endl;
24
25     return 0;
26 }
```

Завдання №5

practice_work_anhelina_bobrynok.cpp

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  void ListBooks(string books[], bool available[]) {
8      for (int i = 0; i < 5; i++) {
9          cout << books[i] << " - " << (available[i] ? "Available" : "Not available") << endl;
10     }
11 }
12
13 void AvailableBooks(string books[], bool available[]) {
14     for (int i = 0; i < 5; i++) {
15         cout << books[i] << " - " << (available[i] ? "Available" : "Not available") << endl;
16     }
17 }
18
19 void TakeBook(string books[], bool available[], vector<string>& users_books) {
20     cout << "Enter the book you would like to take:" << endl;
21     string choose_book;
22     cin.ignore();
23     getline(cin, choose_book);
24     bool found = false;
25
26     for (int i = 0; i < 5; i++) {
27         if (choose_book == books[i] && available[i]) {
28             cout << "You took: " << choose_book << endl;
29             users_books.push_back(choose_book);
30             available[i] = false;
31             found = true;
32             break;
33         }
34     }
35
36     if (!found) {
37         cout << "Sorry, we don't have this book in stock or it's already taken." << endl;
38     }
39 }
40
41 void ReturnBook(string books[], bool availability[], vector<string>& users_books) {
42     cout << "Enter the book you would like to return:" << endl;
43     string choose_book;
44     cin.ignore();
45     getline(cin, choose_book);
46
47     auto it = find(users_books.begin(), users_books.end(), choose_book);
48     if (it != users_books.end()) {
49         for (int i = 0; i < 5; i++) {
50             if (books[i] == choose_book) {
51                 availability[i] = true;
52                 cout << "You returned: " << choose_book << endl;
53                 break;
54             }
55         }
56         users_books.erase(it);
57     } else {
58         cout << "This book is not in your borrowed list." << endl;
59     }
60 }
61
62 int main() {
63     string action;
64     string books[5] = {
65         "Harry Potter and the Philosopher's Stone",
66         "The Little Prince",
67         "Alice's Adventures in Wonderland",
68         "The Hobbit",
69         "Lolita"
70     };
71     bool available[5] = {true, true, true, true, true};
72     vector<string> users_books;
73
74     while (action != "exit") {
75         begin:
76         cout << "\nLibrary Menu\n";
77         cout << "Please enter an action (list, available, take, return, exit): ";
78         cin >> action;
```

```

79
80     if (action == "available") {
81         string continuation;
82         do {
83             AvailableBooks(books, available);
84             cout << "Go to menu?(yes,no): ";
85             cin >> continuation;
86         } while (continuation == "no");
87
88     } else if (action == "take") {
89         string continuation;
90         do {
91             TakeBook(books, available, users_books);
92             cout << "Go to menu?(yes,no): ";
93             cin >> continuation;
94         } while (continuation == "no");
95
96     } else if (action == "list") {
97         string continuation;
98         do {
99             ListBooks(books, available);
100             cout << "Go to menu?(yes,no): ";
101             cin >> continuation;
102         } while (continuation == "no");
103
104     } else if (action == "return") {
105         string continuation;
106         do {
107             ReturnBook(books, available, users_books);
108             cout << "Go to menu?(yes,no): ";
109             cin >> continuation;
110         } while (continuation == "no");
111
112     } else if (action != "exit") {
113         cout << "Invalid action. Try again." << endl;
114         goto begin;
115     }
116 }
117 return 0;
118 }

```

Завдання №6

self_practice_work_anhelina_bobrynok.cpp

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      int n, k;
7      cin >> n >> k;
8      int min_students = k;
9      int max_students = 3 * k;
10
11      if (n < min_students || n > max_students) {
12          cout << "Impossible" << endl;
13          return 0;
14      }
15      vector<int> teams(k, 1);
16      n -= k;
17
18      for (int i = 0; i < k && n > 0; ++i) {
19          int add_students = min(2, n);
20          teams[i] += add_students;
21          n -= add_students;
22      }
23
24      for (int i = 0; i < k; ++i) {
25          cout << teams[i] << " ";
26      }
27      cout << endl;
28
29      return 0;
30 }

```

5)Результати виконання завдань, тестування та фактично затрачений час:

Завдання №1

Сума цілих додатніх непарних чисел, менших 200, дорівнює: 10000

Витрачено часу +-15 хв

Завдання №2

x	S(x) для n	S(x) для ε	Точне значення
0.100000	0.099833	0.099833	0.099833
0.190000	0.188859	0.188859	0.188859
0.280000	0.276356	0.276356	0.276356
0.370000	0.361615	0.361616	0.361615
0.460000	0.443948	0.443948	0.443948
0.550000	0.522687	0.522687	0.522687
0.640000	0.597195	0.597195	0.597195
0.730000	0.666870	0.666869	0.666870
0.820000	0.731146	0.731145	0.731146
0.910000	0.789504	0.789504	0.789504
1.000000	0.841471	0.841471	0.841471

Витрачено часу +-45хв

Завдання №3

Витрачено часу +-45 хв

```
Sum for 5 parameters: 40
Sum for 10 parameters: 330
Sum for 12 parameters: 572
```

Завдання №4

Множення цілих чисел $5 * 4 = 20$

Множення комплексних чисел $(3 + 2i) * (1 + 4i) = (-5, 14)$

Витрачено часу +-20хв

Завдання №5

```
Library Menu
Please enter an action (list, available, take, return, exit): take
Enter the book you would like to take:
The Little Prince
You took: The Little Prince
Go to menu?(yes,no): yes

Library Menu
Please enter an action (list, available, take, return, exit): list
Harry Potter and the Philosopher's Stone - Available
The Little Prince - Not available
Alice's Adventures in Wonderland - Available
The Hobbit - Available
Lolita - Available
Go to menu?(yes,no): yes

Library Menu
Please enter an action (list, available, take, return, exit): return
Enter the book you would like to return:
The Little Prince
You returned: The Little Prince
Go to menu?(yes,no): yes
```

Витрачено часу +-2 години

Завдання №6

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.066	Перегляд

Витрачено часу +-25 хв

Висновки: Завдяки цій лабораторній роботі я мала змогу 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.