

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

Виконала:

Студент групи ШІ-11

Боднар Денис

Тема роботи: Ознайомлення з циклами та їх видами: for, while, do-while, та їх значенням у програмуванні. Вивчення способів управління виконанням циклів за допомогою операторів break і continue, а також передчасного завершення циклів. Аналіз вкладених циклів та їх застосування для вирішення складних завдань. Ознайомлення з основами функцій у C++: визначення, параметри, повернення значень, перевантаження функцій та робота з просторами імен. Вивчення розширених можливостей функцій, таких як еліпсис для змінної кількості параметрів, рекурсія, та використання вбудованих функцій для оптимізації коду.

Мета роботи: Ознайомитися з основними типами циклів у C++ (for, while, do-while), їхньою роллю та синтаксисом, а також навчитися застосовувати їх у різних завданнях. Навчитися використовувати оператори break та continue, розуміти умови завершення та передчасне завершення циклів. Засвоїти принципи створення функцій, включаючи передавання параметрів, значення за замовчуванням та повернення результатів. Розібратися з поняттям перевантаження функцій та використанням просторів імен для організації коду. Опанувати роботу з функціями зі змінною кількістю параметрів, а також принципами рекурсії та їхнього застосування у складних задачах. Ознайомитися з вбудованими функціями у C++ та їх роллю у спрощенні коду й виконанні стандартних операцій. Навчитися використовувати вкладені цикли та практично застосовувати їх для вирішення комплексних завдань у програмуванні.

Теоретичні відомості:

1. Теоретичні відомості з переліком важливих тем:

- Тема №1: Введення в цикли та їх види в C++. Вкладені цикли.
- Тема № 2: Управління виконанням циклів.
- Тема № 3: Основи Функцій у C++.
- Тема № 4: Перевантаження Функцій та Простір Імен.
- Тема № 5: Функції зі змінною кількістю параметрів (еліпсис).
- Тема № 6: Рекурсія: основи, приклади рекурсивних функцій та їх аналіз.
- Тема № 7: Вбудовані Функції в C++.

2. Індивідуальний план опрацювання теорії:

Тема №1: Введення в цикли та їх види в C++. Вкладені цикли.

- Джерела:
 - <https://acode.com.ua/urok-72-tsykl-for>
 - <https://acode.com.ua/urok-70-tsykl-while/>

- <https://acode.com.ua/urok-69-operator-goto/>
- Що опрацьовано:
 - Область видимості циклу.
 - Помилка неврахованої одиниці.
 - Пропуск виразів в циклі for та запис в форматі for (; i<10;)
 - Тип signed int для лічильників циклу.
- Статус: Ознайомлений

Тема №2: Управління виконанням циклів.

- Джерела:
 - <https://acode.com.ua/urok-69-operator-goto/>
- Що опрацьовано:
 - Нескінченний цикл while та return, break, exit, goto для виходу з нього.
 - Оператор goto та лейбл.
- Статус: Ознайомлений

Тема №3: Основи Функцій у C++.

- Джерела:
 - <https://acode.com.ua/urok-15-funktsiyi-i-operator-return/>
 - <https://www.youtube.com/watch?v=VKQ242d-Rag>
- Що опрацьовано:
 - Особливості функцій типу void та їх (виклику) виводу.
 - Переривання роботи функції.
 - Скелет(прототип) функції.
- Статус: Ознайомлений

Тема №4: Перевантаження Функцій та Простір Імен.

- Джерела:
 - <https://www.youtube.com/watch?v=kYxNioENAIo>
 - <https://www.youtube.com/watch?v=3KJfisev6SI>
- Що опрацьовано:
 - Переваги перевантаження функцій.
 - Перевантаження зміною кількості аргументів.
 - Перевантаження зміною типів аргументів.
 - Перевантаження cout.
- Статус: Ознайомлений

Тема №5: Функції зі змінною кількістю параметрів (еліпсис).

- Джерела:
 - <https://acode.com.ua/urok-117-elipsys/>
- Що опрацьовано:
 - доступ до еліпсису через va_list.

- Ініціалізуємо `va_list`, використовуючи `va_start`. Перший параметр - це список, який потрібно ініціалізувати. Другий параметр - це останній параметр, який не є еліпсисом.
- Використовуємо `va_arg` для отримання параметрів з еліпсиса. Перший параметр - це `va_list`, який ми використовуємо. Другий параметр - це очікуваний тип параметрів.
- Виконуємо очищення `va_list`, коли вже зробили все необхідне за допомогою `va_end`.
- Параметри еліпсиса не мають оголошень типу даних. При їх використанні компілятор повністю пропускає перевірку типів даних.
- Статус: Ознайомлений

Тема №6: Рекурсія: основи, приклади рекурсивних функцій та їх аналіз.

- Джерела:
 - https://www.youtube.com/watch?v=V7q9w_s0nns&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=54&t=31s
- Що опрацьовано:
 - Рекурсія та її завдання.
 - Недолік рекурсії щодо використання пам'яті через постійне створення змінних.
- Статус: Ознайомлений

Тема №7: Вбудовані Функції в C++.

- Джерела:
 - <https://www.youtube.com/watch?v=ZUrdKTBRXWo>
- Що опрацьовано:
 - Для чого вбудовувати функції.
 - В яких випадках функції не можуть бути вбудованими.
- Статус: Ознайомлений

Виконання роботи:

Завдання №1 VNS Lab 2 - Task 1-13

13) Знайти суму ряду з точністю $\epsilon=0.0001$, загальний член якого

$$a_n = \frac{3^n n!}{(3n)!}$$

Завдання №2 VNS Lab 3 - Task 1-13

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 ($\varepsilon=0.0001$).

Для порівняння знайти точне значення функції.

13	$y = \frac{1}{2} \ln x$	$0,2 \leq x \leq 1$	10	$S = \frac{x-1}{x+1} + \frac{1}{3} \left(\frac{x-1}{x+1} \right)^3 + \dots + \frac{1}{2n+1} \left(\frac{x-1}{x+1} \right)^{2n+1}$
----	-------------------------	---------------------	----	---

1. Алгоритм розв'язання завдання зводиться до трьох циклів, причому два з них вкладені в третій. Внутрішні цикли підсумують доданки при фіксованому параметрі x , один (арифметичний для заданого n), інший (ітераційний для заданої точності). При організації цих циклів варто звернути увагу на правильний вибір формули для обчислення елемента ряду a_n і правильне присвоєння початкових значень змінним циклу. Зовнішній цикл організує зміну параметра x .

2. Результати розрахунків надрукувати у такому вигляді:

Обчислення функції

X=.....	SN=.....	SE=.....	Y=.....
X=.....	SN=.....	SE=.....	Y=.....
.....			
X=.....	SN=.....	SE=.....	Y=.....

Тут X - значення параметра; SN - значення суми для заданого n ; SE - значення суми для заданої точності; Y -точне значення функції.

Завдання №3 VNS Lab 7 - Task 1-13

Написати функцію зі змінною кількістю параметрів для перекладу чисел з десяткової системи числення в трійкову. Написати викликаючу функцію `main`, що звертається до цієї функції не менше трьох разів з кількістю параметрів 3, 4, 7.

Завдання №4 VNS Lab 7 - Task 1-13

Написати перевантажені функції й основну програму, що їх викликає.

а) для перетворення десяткового дробу у звичайний;

б) для перетворення звичайного дробу у десятковий.

Завдання №5 Class 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.

Завдання №6 Self Practice Work

Найбільша зростаюча підпоследовність

Вам задано послідовність із n цілих чисел a_i . Ваша задача — знайти довжину найбільшої зростаючої підпослідовності заданої послідовності.

Вхідні дані

У першому рядку задано ціле число n .

У наступному рядку задано n цілих чисел — послідовність a_i .

Вихідні дані

Виведіть довжину найбільшої зростаючої підпослідовності.

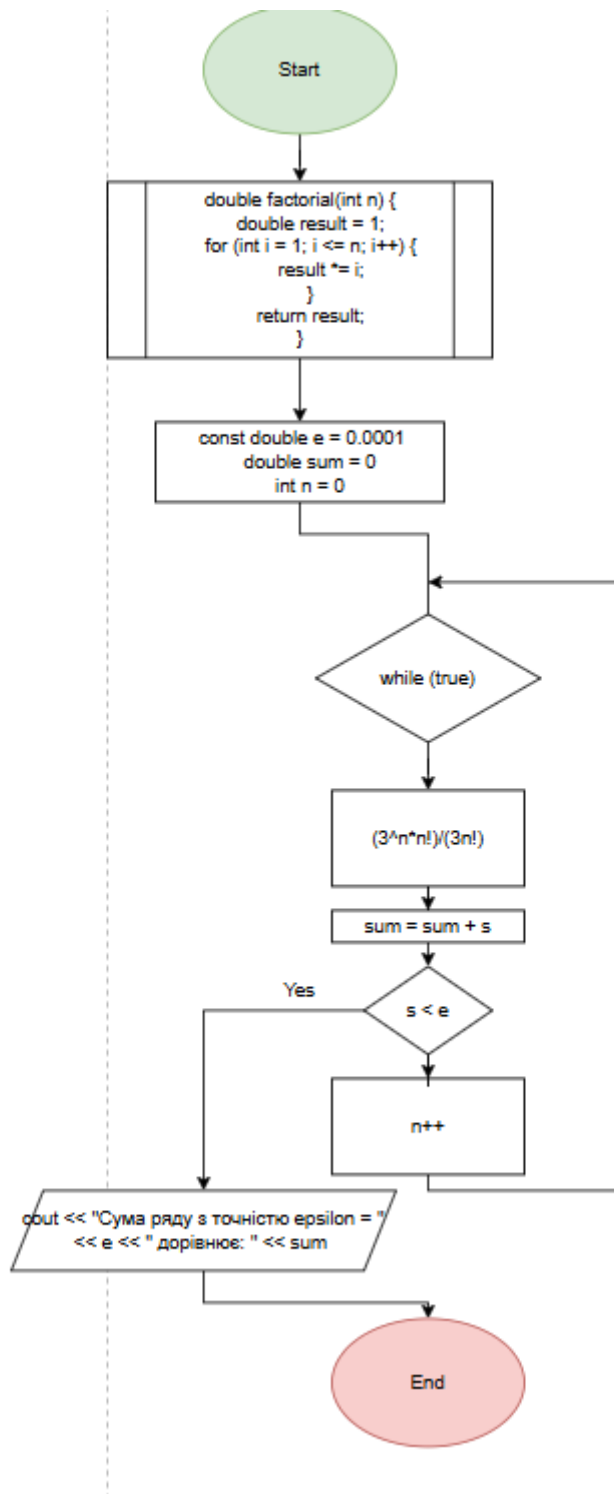
Обмеження

$$1 \leq n \leq 100,$$

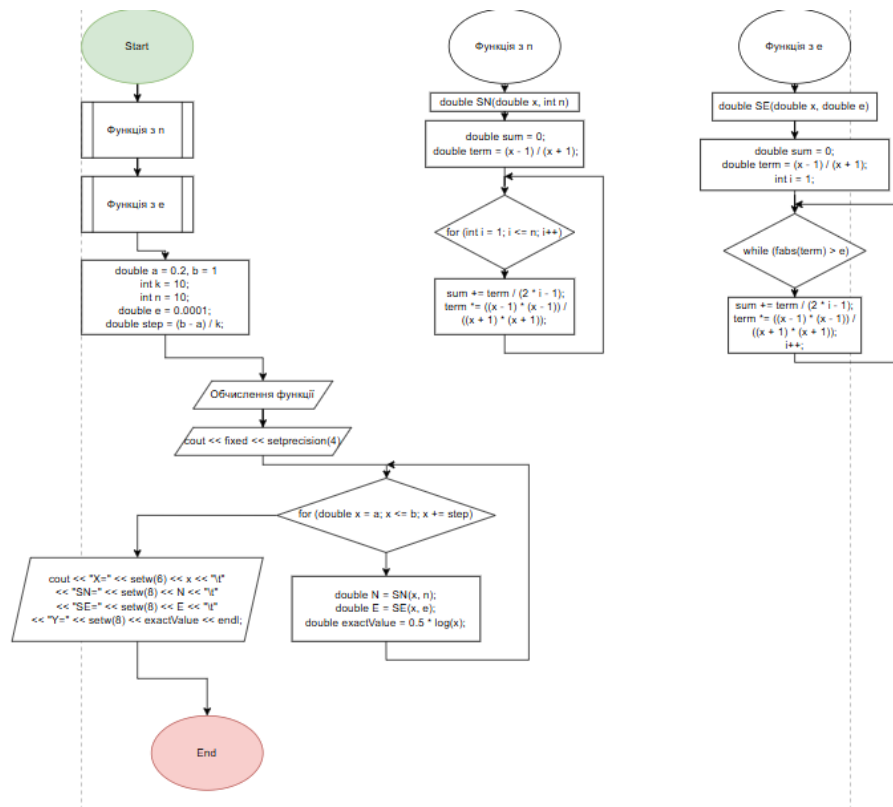
$$1 \leq a_i \leq 10^9.$$

Дизайн та планована оцінка часу виконання завдань:

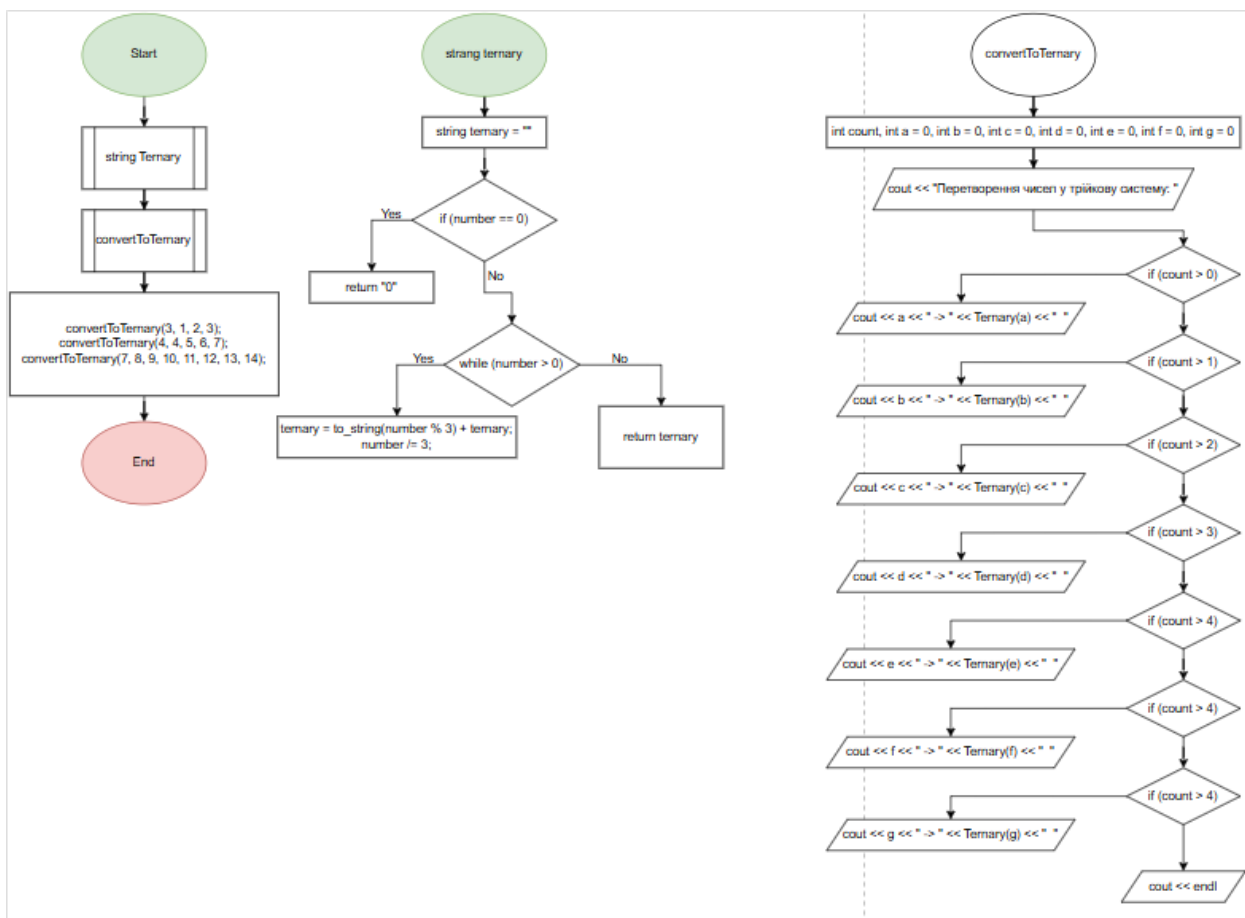
Завдання №1 VNS Lab 2 - Task 1-13



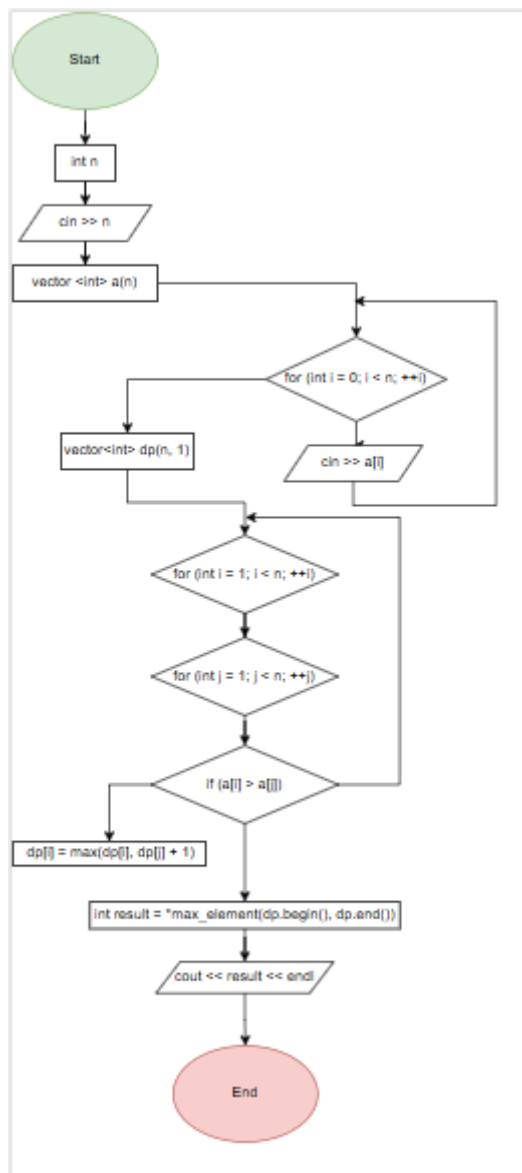
Завдання №2 VNS Lab 3 - Task 1-13



Завдання №3 VNS Lab 7 - Task 1-13



Завдання №4 VNS Lab 7 - Task 2-13



Код програм з посиланням на зовнішні ресурси

Завдання №1

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  double factorial(int n) {
7      double result = 1;
8      for (int i = 1; i <= n; i++) {
9          result *= i;
10     }
11     return result;
12 }
13
14 int main() {
15     const double e = 0.0001;
16     double sum = 0;
17     int n = 0;
18
19     while (true) {
20         double s = pow(3, n) * factorial(n) / factorial(3 * n);
21
22         sum = sum + s;
23
24         if (s < e) {
25             break;
26         }
27
28         n++;
29     }
30
31     cout << "Сума ряду з точністю epsilon = " << e << " дорівнює: " << sum << endl;
32
33     return 0;
34 }

```

Завдання №2

```

1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5
6  // Функція з n
7  double SN(double x, int n) {
8      double sum = 0;
9      double term = (x - 1) / (x + 1);
10     for (int i = 1; i <= n; i++) {
11         sum += term / (2 * i - 1);
12         term *= ((x - 1) * (x - 1)) / ((x + 1) * (x + 1));
13     }
14     return sum;
15 }
16
17 // Функція з e
18 double SE(double x, double e) {
19     double sum = 0;
20     double term = (x - 1) / (x + 1);
21     int i = 1;
22     while (fabs(term) > e) {
23         sum += term / (2 * i - 1);
24         term *= ((x - 1) * (x - 1)) / ((x + 1) * (x + 1));
25         i++;
26     }
27     return sum;
28 }
29
30 int main() {
31     double a = 0.2, b = 1;
32     int k = 10;
33     int n = 10;
34     double e = 0.0001;
35     double step = (b - a) / k;
36
37     cout << "Обчислення функції\n";
38     cout << fixed << setprecision(4); // встановлюємо точність для чисел
39     for (double x = a; x <= b; x += step) {
40         double N = SN(x, n);
41         double E = SE(x, e);
42         double exactValue = 0.5 * log(x);
43
44         cout << "X=" << setw(6) << x << "\t"
45             << "SN=" << setw(8) << N << "\t"
46             << "SE=" << setw(8) << E << "\t"
47             << "Y=" << setw(8) << exactValue << endl;
48     }
49
50     return 0;
51 }

```

Завдання №3

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  string Ternary(int number) {
7      string ternary = "";
8      if (number == 0) return "0";
9
10     while (number > 0) {
11         ternary = to_string(number % 3) + ternary;
12         number /= 3;
13     }
14     return ternary;
15 }
16
17 void convertToTernary(int count, int a = 0, int b = 0, int c = 0, int d = 0, int e = 0, int f = 0, int g = 0) {
18     cout << "Перетворення чисел в трійкову систему: ";
19     if (count > 0) cout << a << " -> " << Ternary(a) << " ";
20     if (count > 1) cout << b << " -> " << Ternary(b) << " ";
21     if (count > 2) cout << c << " -> " << Ternary(c) << " ";
22     if (count > 3) cout << d << " -> " << Ternary(d) << " ";
23     if (count > 4) cout << e << " -> " << Ternary(e) << " ";
24     if (count > 5) cout << f << " -> " << Ternary(f) << " ";
25     if (count > 6) cout << g << " -> " << Ternary(g) << " ";
26     cout << endl;
27 }
28
29 int main() {
30     convertToTernary(3, 0, 2, 3);
31     convertToTernary(4, 4, 5, 6, 7);
32     convertToTernary(7, 8, 9, 10, 11, 12, 13, 14);
33     return 0;
34 }

```

Завдання №4

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  void drib10(double decimal) {
7      int precision = 1000000; //висока точність для перетворення
8      int znam = precision;
9      int chyseln = round(decimal * precision);
10
11     int nsd = 1;
12     for (int i = 1; i <= chyseln && i <= znam; i++) {
13         if (chyseln % i == 0 && znam % i == 0) {
14             nsd = i;
15         }
16     }
17
18     chyseln /= nsd;
19     znam /= nsd;
20
21     cout << "Звичайний дріб: " << chyseln << "/" << znam << endl;
22 }
23
24 double prostdribv10(int chyseln, int znam, int precision = 1) {
25     double result = static_cast<double>(chyseln) / znam;
26     return round(result * precision) / precision;
27 }
28
29 int main() {
30     double decimal = 0.77;
31     cout << "Десятковий дріб в звичайний: ";
32     drib10(decimal);
33
34     cout << "Звичайний дріб в десятковий: ";
35     cout << prostdribv10(6, 7) << endl;
36
37     cout << "Звичайний дріб в десятковий з точною точністю: ";
38     cout << prostdribv10(1, 2, 3000) << endl;
39
40     return 0;
41 }

```

Завдання №5

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  const int numBooks = 4;
7  string books[numBooks] = {"1984", "Бідний тато, багатий тато", "Код да Вінчі", "Біблія"};
8  bool Borrowed[numBooks] = {false, false, false, false};
9
10 void listBooks() {
11     cout << "\nПерелік книг:\n";
12     for (int i = 0; i < numBooks; i++) {
13         cout << (i + 1) << ". " << books[i] << (Borrowed[i] ? " (зайнята)" : " (доступна)") << endl;
14     }
15 }
16
17 bool isBookAvailable(int index) {
18     if (index < 1 || index > numBooks) {
19         cout << "Помилка: Неправильний номер книги.\n";
20         return false;
21     }
22     return true;
23 }
24
25 bool borrowBook() {
26     int bookIndex;
27     cout << "Введіть номер книги, яку хочете взяти: ";
28     cin >> bookIndex;
29
30     if (!isBookAvailable(bookIndex)) {
31         return true;
32     }
33
34     if (Borrowed[bookIndex - 1]) {
35         cout << "На жаль, книга \"" << books[bookIndex - 1] << "\" вже зайнята.\n";
36     } else {
37         Borrowed[bookIndex - 1] = true;
38         cout << "Ви успішно взяли книгу \"" << books[bookIndex - 1] << "\".\n";
39     }
40     return false;
41 }
42
43 bool returnBook() {
44     int bookIndex;
45     cout << "Введіть номер книги, яку хочете повернути: ";
46     cin >> bookIndex;
47
48     if (!isBookAvailable(bookIndex)) {
49         return true;
50     }
51
52     if (!Borrowed[bookIndex - 1]) {
53         cout << "Ця книга вже є в бібліотеці.\n";
54     } else {
55         Borrowed[bookIndex - 1] = false;
56         cout << "Ви успішно повернули книгу \"" << books[bookIndex - 1] << "\".\n";
57     }
58     return false;
59 }
```

```

61 int main() {
62     int choice;
63     bool Uexit = false;
64
65     do {
66         main_menu:
67         cout << "\nГоловне меню:\n";
68         cout << "1. Переглянути всі книги\n";
69         cout << "2. Взяти книгу\n";
70         cout << "3. Повернути книгу\n";
71         cout << "4. Вийти\n";
72         cout << "Виберіть дію: ";
73         cin >> choice;
74
75         switch (choice) {
76             case 1:
77                 listBooks();
78                 break;
79             case 2:
80                 if (borrowBook()) {
81                     goto main_menu;
82                 }
83                 break;
84             case 3:
85                 if (returnBook()) {
86                     goto main_menu;
87                 }
88                 break;
89             case 4:
90                 Uexit = true;
91                 cout << "Дякуємо за користування бібліотекою!\n";
92                 break;
93             default:
94                 cout << "Неправильний вибір! Спробуйте знову.\n";
95                 goto main_menu;
96         }
97
98         if (!Uexit) {
99             string usercin;
100             cout << "Бажаєте перейти до головного меню? (Yes/Exit): ";
101             cin >> usercin;
102             if (usercin == "exit" || usercin == "Exit") {
103                 Uexit = true;
104                 cout << "Дякуємо за користування бібліотекою!\n";
105             } else if (usercin == "yes" || usercin == "Yes") {
106                 goto main_menu;
107             } else {
108                 cout << "Неправильний вибір, спробуйте ще раз\n";
109             }
110         }
111     } while (!Uexit);
112
113     return 0;
114 }

```

Завдання №6


```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main() {
8      int n;
9      cin >> n;
10
11     vector<int> a(n);
12     for (int i = 0; i < n; ++i) {
13         cin >> a[i];
14     }
15
16     vector<int> dp(n, 1);
17
18     for (int i = 1; i < n; ++i) {
19         for (int j = 0; j < i; ++j) {
20             if (a[i] > a[j]) {
21                 dp[i] = max(dp[i], dp[j] + 1);
22             }
23         }
24     }
25
26     int result = *max_element(dp.begin(), dp.end());
27     cout << result << endl;
28
29     return 0;
30 }

```

Результати виконання завдань, тестування та фактично затрачений час:

Завдання №1 VNS Lab 2 - Task 1-13

Сума ряду з точністю epsilon = 0.0001 дорівнює: 1.52545

Фактично затрачений час: 40хв

Завдання №2 VNS Lab 3 - Task 1-13

Обчислення функції			
X=0.2000	SN= -0.8047	SE= -0.8047	Y= -0.8047
X=0.2800	SN= -0.6365	SE= -0.6365	Y= -0.6365
X=0.3600	SN= -0.5108	SE= -0.5108	Y= -0.5108
X=0.4400	SN= -0.4105	SE= -0.4105	Y= -0.4105
X=0.5200	SN= -0.3270	SE= -0.3270	Y= -0.3270
X=0.6000	SN= -0.2554	SE= -0.2554	Y= -0.2554
X=0.6800	SN= -0.1928	SE= -0.1928	Y= -0.1928
X=0.7600	SN= -0.1372	SE= -0.1372	Y= -0.1372
X=0.8400	SN= -0.0872	SE= -0.0872	Y= -0.0872
X=0.9200	SN= -0.0417	SE= -0.0417	Y= -0.0417
X=1.0000	SN= -0.0000	SE= 0.0000	Y= -0.0000

Фактично затрачений час: 1год

Завдання №3 VNS Lab 7 - Task 1-13

```

Перетворення чисел у трійкову систему: 0 -> 0  2 -> 2  3 -> 10
Перетворення чисел у трійкову систему: 4 -> 11  5 -> 12  6 -> 20  7 -> 21
Перетворення чисел у трійкову систему: 8 -> 22  9 -> 100  10 -> 101  11 -> 102  12 -> 110  13 -> 111  14 -> 112

```

Фактично затрачений час: 1,5год

Завдання №4 VNS Lab 7 - Task 2-13

```

Десятковий дріб у звичайний: Звичайний дріб: 77/100
Звичайний дріб у десятковий: 1
Звичайний дріб у десятковий з точною точністю: 0.5

```

Фактично затрачений час: 45хв

Завдання №5 Class Practice Work

```

Головне меню:
1. Переглянути всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти
Виберіть дію: 1

Список книг:
1. 1984 (доступна)
2. Бідний тато, багатий тато (доступна)
3. Код да Вінчі (доступна)
4. Біблія (доступна)
Бажаєте перейти до головного меню? (Yes/Exit): yes

Головне меню:
1. Переглянути всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти
Виберіть дію: 2
Введіть номер книги, яку хочете взяти: 1
Ви успішно взяли книгу "1984".
Бажаєте перейти до головного меню? (Yes/Exit): yes

Головне меню:
1. Переглянути всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти
Виберіть дію: 3
Введіть номер книги, яку хочете повернути: 1
Ви успішно повернули книгу "1984".
Бажаєте перейти до головного меню? (Yes/Exit): exit
Дякуємо за користування бібліотекою!

```

Фактично затрачений час: 3год

Завдання №6 Self Practice Work

```

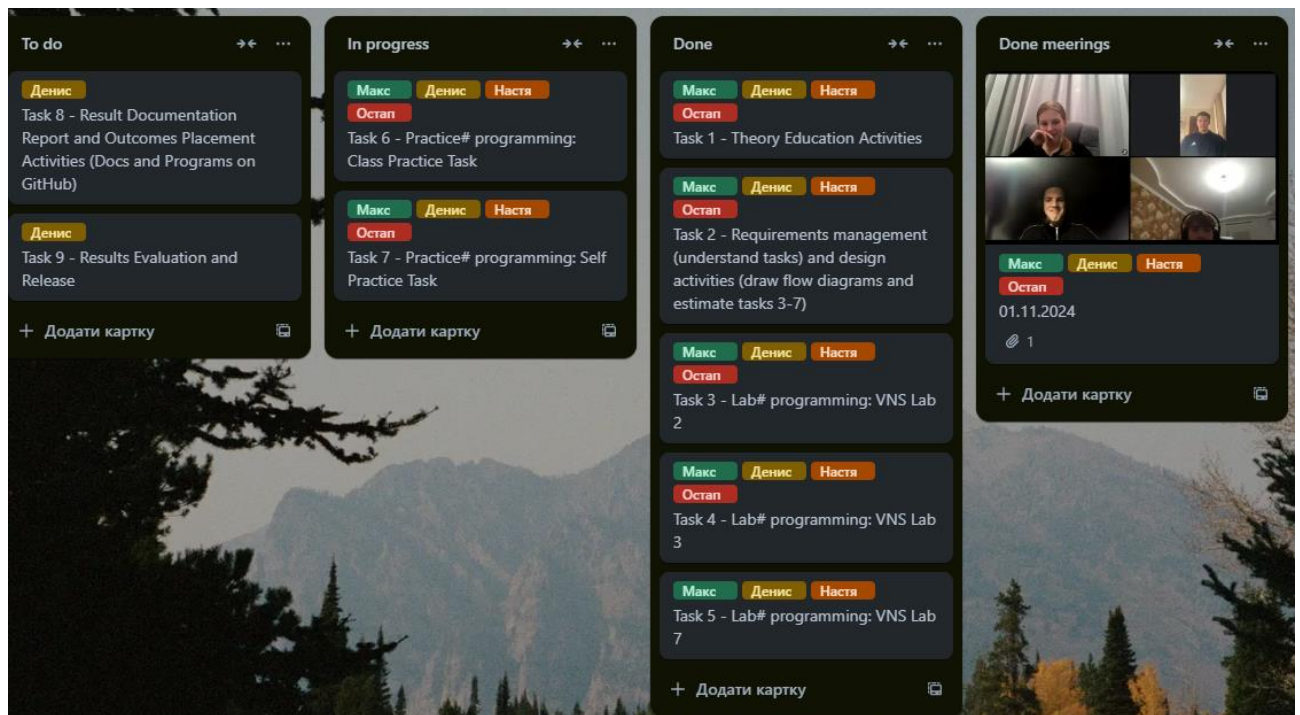
5
4 7 2 10 3
3

```

Створено	Задача	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	№
5 годин тому	0002 - Найбільша зростаюча підпоследовність	C++ 23	Зарховано	0.003	1.188	1860639

Фактично затрачений час: 30хв

Кооперація з командою:



Ми зустрічалися один раз з командою, на цій зустрічі кожен розпитав, що не розуміє ми один одному помогли а також розставили пріоритети в дошці трелло

Висновок:

Виконуючи третій епік, я ознайомився з різними видами циклів у C++ і зрозумів, як вони можуть спростити реалізацію повторюваних завдань у програмуванні. Навчився керувати виконанням циклів за допомогою операторів `break` і `continue`, що дозволяє змінювати логіку їх роботи. Дослідження функцій, їх параметрів і перевантаження допомогло мені краще організовувати код і робити його більш зрозумілим. Також я опанував застосування рекурсії та функцій зі змінною кількістю параметрів, що розширило моє уявлення про можливості функцій у складних завданнях. Завдяки вивченню вбудованих функцій, я побачив, як C++ може оптимізувати виконання стандартних операцій і полегшити написання коду. Ці знання я закріпив на практиці, написавши кілька програм.