

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(ла):**  
Студент групи ІІІ-11  
Зубрицький Арсеній Юрійович

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 працювати з функціями, циклами та перевантаженими функціями. Застосувати на практиці вивчений теоретичний матеріал.

## **Теоретичні відомості:**

1) Теоретичні відомості з переліком важливих тем:

- **Тема №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.
- **Тема №2:** Управління Виконанням Циклів:
  - Застосування операторів break та continue.
  - Умови завершення циклів.
  - Передчасне завершення виконання циклу.
  - Приклади та вправи з управлінням циклами.
- **Тема №3:** Вкладені Цикли:
  - Поняття та важливість вкладених циклів.
  - Реалізація вкладених циклів: приклади для різних сценаріїв.
  - Практичні завдання на вкладені цикли.
- **Тема №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.
- **Тема №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.
- **Тема №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.

- **Тема №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.
- 2) Індивідуальний план опрацювання теорії:
- **Тема №1:** Введення в Цикли та їх Види в C++
  - Джерела Інформації
    - [https://www.youtube.com/watch?v=vLnPwxZdW4Y&list=PLWKjhJtqVAbmUE5IqyfGYEYjrZBYzaT4m&index=3&ab\\_channel=freeCodeCamp.org](https://www.youtube.com/watch?v=vLnPwxZdW4Y&list=PLWKjhJtqVAbmUE5IqyfGYEYjrZBYzaT4m&index=3&ab_channel=freeCodeCamp.org)
    - <https://acode.com.ua/urok-66-operator-y-upravlinnya-potokom-vykonnannya-program/#toc-4>
    - <https://acode.com.ua/urok-70-tsykl-while/>
    - <https://acode.com.ua/urok-71-tsykl-do-while/>
    - <https://acode.com.ua/urok-72-tsykl-for/>
  - Що опрацьовано:
    - Ознайомився з основними циклами в C++ та прикладами застосування в програмуванні
    - Зрозумів синтаксис циклів
  - Статус: Ознайомлений
  - Початок опрацювання теми: 15.11.2024
  - Звершення опрацювання теми: 15.11.2024
  - Витрачено часу: 30 хв
- **Тема №2:** Управління Виконанням Циклів
  - Джерела Інформації:
    - <https://acode.com.ua/urok-73-operator-break-i-continue/>
  - Що опрацьовано:
    - Ознайомився з операторами break і continue, та оглянув практичне застосування операторів на практиці
    - На практиці застосував у коді
  - Статус: Ознайомлений
  - Початок опрацювання теми: 15.11.2024
  - Звершення опрацювання теми: 15.11.2024
  - Витрачено часу: 30 хв
- **Тема №3:** Вкладені Цикли
  - Джерела Інформації:
    - <https://acode.com.ua/urok-72-tsykl-for/>
    - <https://acode.com.ua/urok-70-tsykl-while/>
  - Що опрацьовано:
    - Зрозумів суть вкладених циклів
    - Та на практиці закріпив їх використання в різних задачах
  - Статус: Ознайомлений
  - Початок опрацювання теми: 15.11.2024

- Звершення опрацювання теми: 15.11.2024
- Витрачено часу: 28 хв
- **Тема №4: Основи Функцій у C++**
  - Джерела Інформації:
    - <https://acode.com.ua/urok-15-funktsiyi-i-operator-return/>
    - <https://youtu.be/vLnPwxZdW4Y?list=PLWKjhJtqVAbmUE5IqyfGYEYjrZBYzaT4m>
  - Що опрацьовано:
    - Ознайомився з поняттям функції, оголошенням її
    - На практиці реалізував застосування декількох функцій
  - Статус: Ознайомлений
  - Початок опрацювання теми: 15.11.2024
  - Звершення опрацювання теми: 15.11.2024
  - Витрачено часу: 36 хв
- **Тема №5: Перевантаження Функцій та Простір Імен**
  - Джерела Інформації:
    - <https://youtu.be/kYxNioENAIo>
    - <https://acode.com.ua/urok-108-perevantazhennya-funktsij/>
  - Що опрацьовано:
    - зрозумів концепцію використання перевантаження функції
    - ознайомився з поняттям та на прикладах із застосуванням просторів імен
  - Статус: Ознайомлений
  - Початок опрацювання теми: 15.11.2024
  - Звершення опрацювання теми: 15.11.2024
  - Витрачено часу: 30 хв
- **Тема №6: Розширені Можливості Функцій**
  - Джерела Інформації:
    - Лекції
    - <https://acode.com.ua/urok-117-elipsys/>
    - <https://acode.com.ua/urok-52-globalni-zminni/>
    - <https://acode.com.ua/urok-113-rekursiya-i-chysla-fibonachchi/>
    - <https://acode.com.ua/urok-86-vkazivnyky-i-masyvy/>
  - Що опрацьовано:
    - Ознайомився з еліпсисом та практичним застосуванням його.
    - Ознайомився з областями видимості змінних та ключовими словами static, extern.
    - Дослідив рекурсивні функції та їх практичне застосування.
  - Статус: Ознайомлений
  - Початок опрацювання теми: 15.11.2024
  - Звершення опрацювання теми: 15.11.2024
  - Витрачено часу: 30 хв
- **Тема №7: Вбудовані Функції в C++**
  - Джерела Інформації:

- <https://acode.com.ua/urok-107-vbudovani-funktsiyi/>
- 
- Що опрацьовано:
  - Ознайомився з функціями printf() и scanf()
  - Написав лінійний алгоритм та алгоритми з використанням оператора switch
- Статус: Ознайомлений
- Початок опрацювання теми: 15.11.2024
- Звершення опрацювання теми: 15.11.2024
- Витрачено часу: 30 хв

## Виконання роботи:

### 1. Опрацювання завдання та вимог до програм та середовища:

#### Завдання №1 Vns Lab\_2\_task\_1\_25

25) Знайти суму 7 членів ряду, у якому

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

#### - Вимоги:

1. При визначенні суми членів ряду варто використовувати рекурентну формулу для отримання наступного члена ряду.

Наприклад, потрібно знайти суму ряду з точністю  $\varepsilon=0.0001$ , загальний член

якого 
$$a_n = \frac{2(n!)^2}{(3(2n)!)!}.$$

Для одержання рекурентної формули обчислимо відношення:

$$\frac{a_{n+1}}{a_n} = \frac{2((n+1)!)^2 \cdot 3(2n)!}{3(2n+2)! \cdot 2(n!)^2} = \frac{n+1}{2(2n+1)},$$

звідки:

$$a_{n+1} = a_n \cdot \frac{(n+1)}{2(2n+1)}.$$

2. При складанні програми вважати, що точність досягнута, якщо  $a_n < \varepsilon$

#### Завдання №2 VNS Lab\_3\_task\_1\_25

25	$y = \frac{e^x - e^{-x}}{2}$	$0,1 \leq x \leq 1$	20	$S = x + \frac{x^3}{3!} + \dots + \frac{x^{2n+1}}{(2n+1)!}$
----	------------------------------	---------------------	----	---

- 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.
- **Вимоги:** 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  $x$ .

- 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\_25

- **Умова:** Написати функцію (або макровизначення), що знаходить довжину сторони по координатах його точок.. Написати функцію square, що обчислює площу трикутника, заданого координатами вершин. Написати функцію square 1 з змінною кількістю параметрів, що визначає площу трикутника, що містить діагональ найбільшої довжини опуклого багатокутника, заданого координатами своїх вершин.
- **Вимоги:** Розв'язати зазначене у варіанті завдання, використовуючи функції зі змінною кількістю параметрів.

#### Завдання №4 VNS Lab\_7\_task\_2\_25

- **Умова:** а) для двовимірного масиву знищує всі парні рядки; б) для одновимірного масиву знищує всі елементи, що містяться між двома нульовими елементами
- **Вимоги:** Написати перевантажені функції й основну програму, що їх викликає.

#### Завдання №5 Менеджмент бібліотеки

## Задача

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:

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 Algotester

### Офісна Вулиця. Частина 1

Limits: 2 sec., 256 MiB

Зустрілися якось працівники великих компаній і почали... Обговорювати план вулиці.

Виявляється, всі приміщення, які орендуватимуть ці компанії, збудують вздовж однієї вулиці.

$i$ -та компанія орендуватиме офіс довжиною  $l_i$  метрів. Офіси будуватимуть один за одним, починаючи з точки 0. Всі працівники приїжджатимуть на стоянку, яку побудують в точці 0, та будуть йти до офісів своїх компаній.

Тобто, якщо офіси будуть збудовані в порядку  $p_1, p_2, \dots, p_n$ , то перший офіс почнеться в точці 0 і закінчиться в точці  $l_{p_1}$ , другий почнеться в  $l_{p_1}$  і закінчиться в  $l_{p_1} + l_{p_2}$  і т.д. Двері кожного офісу завжди є в кінці будинку, який є ближчим до стоянки.

Ваше завдання — допомогти розмістити офіси компаній на цій вулиці в такому порядку, щоб сумарна відстань від точки 0 до усіх офісів була мінімальною.

#### Input

У першому рядку задане ціле число  $n$  — кількість компаній.

У наступному рядку задано  $n$  цілих чисел  $l_i$  через пробіл — довжини офісів усіх компаній.

#### Output

У єдиному рядку введіть  $n$  чисел від 1 до  $n$  — порядок компаній, в якому варто будувати офіси.

Якщо існує декілька оптимальних порядків — введіть будь-який із них.

## 2. Дизайн та планована оцінка часу виконання завдань:

### Програма №1 Vns Lab\_2\_task\_1\_25

- Орієнтований час на реалізацію: 30 хв

### Програма №2 VNS Lab\_3\_task\_1\_25

- Орієнтований час на реалізацію: 1 година

### Програма №3 Lab\_7\_task\_1\_25

- Запланований час на реалізацію: 30 хв

### Програма №4 Lab\_7\_task\_2\_25

- Запланований час на реалізацію: 30 хв

### Програма №5 Менеджмент бібліотеки

- Запланований час на реалізацію: 1.5 години

### Програма №6 Self practice Algotester

- Запланований час на реалізацію: 30 хв

## 4. Код програм з посиланням на зовнішні ресурси:

### Завдання №1



```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  // Обчислення факторіала
6  int factorial(int n) {
7      if (n = 1) return 1; // Базові випадки
8      else if(n = 0)return 0;
9      return n * factorial(n - 1);
10 }
11
12 // Обчислення члену ряду за рекурентною формулою
13 double calculateNext(double current, int n) {
14     return current * (n + 1) / (2.0 * (2 * n + 1)); // Рекурентна формула
15 }
16
17 int main() {
18     const int terms = 7;
19     double sum = 0.0;
20     double current = 1.0;
21
22     // Обчислення суми
23     for (int n = 1; n <= terms; n++) {
24         current = pow(n, 2) * exp(-sqrt(n));
25         sum += current;
26     }
27
28     cout << "Сума " << terms << " членів ряду: " << sum << endl;
29
30     return 0;
31 }
32

```

Посилання на pull-request:

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/658/files#diff-466ba368741a6f2a877d52f86f98f7146eec004e7a14a66b6f025dcb63ac6169](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/658/files#diff-466ba368741a6f2a877d52f86f98f7146eec004e7a14a66b6f025dcb63ac6169)

Завдання №2

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  // Точне значення функції
6  double exact_value(double x) {
7      return (exp(x) - exp(-x)) / 2;
8  }
9
10 // Рекурсивна функція для обчислення факторіала
11 int factorial(int n) {
12     if (n <= 1) return 1;
13     return n * factorial(n - 1);
14 }
15
16 // Функція для обчислення SN для заданого числа членів ряду n
17 double sum_series_n(double x, int n) {
18     double sum = 0.0;
19     for (int i = 0; i <= n; i++) {
20         double term = pow(x, 2 * i + 1) / factorial(2 * i + 1);
21         sum += term;
22     }
23     return sum;
24 }

```

```

26 // Функція для обчислення SN для заданої точності epsilon
27 double sum_series_epsilon(double x, double epsilon) {
28     double sum = 0.0;
29     double term;
30     int n = 0;
31     do {
32         term = pow(x, 2 * n + 1) / factorial(2 * n + 1);
33         sum += term;
34         n++;
35     } while (fabs(term) > epsilon);
36     return sum;
37 }
38
39 int main() {
40     double a = 0.1, b = 1.0;
41     int k = 10;
42     double epsilon = 0.0001;
43     int n = 20; // Кількість членів ряду для часткового обчислення
44
45     // Зміна параметра x
46     for (double x = a; x <= b; x += (b - a) / k) {
47         // Обчислення SN для заданого n
48         double sn_n = sum_series_n(x, n);
49         // Обчислення SN для заданої точності epsilon
50         double sn_epsilon = sum_series_epsilon(x, epsilon);
51         // Точне значення
52         double exact = exact_value(x);
53
54         // Виведення результатів
55         cout << "X = " << x << " SN = " << sn_n << " SE = " << sn_epsilon << " Y = " << exact << endl;
56     }
57
58     return 0;
59 }
60

```

Посилання на pull-request:

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/658/files#diff-4f132896de5345179e071630282070aa91f17eb78ab85e55fb191c48db7a93f3](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/658/files#diff-4f132896de5345179e071630282070aa91f17eb78ab85e55fb191c48db7a93f3)

## Завдання №3

```
1  #include <iostream>
2  #include <cmath>
3  #include <cstdarg>
4
5  using namespace std;
6
7  // Функція для обчислення довжини сторони за координатами її кінцевих точок
8  double sideLength(double x1, double y1, double x2, double y2) {
9      return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));
10 }
11
12 // Функція для обчислення площі трикутника за координатами його вершин
13 double square(double x1, double y1, double x2, double y2, double x3, double y3) {
14     return 0.5 * abs(x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2));
15 }
16
17 // Функція для обчислення площі трикутника, що містить найдовшу діагональ в опуклому багатокутнику
18 // зі змінною кількістю вершин (потрібно передати щонайменше 6 параметрів для 3 вершин)
19 double square1(int numPoints, ...) {
20     if (numPoints < 6 || numPoints % 2 != 0) {
21         cerr << "Невірна кількість координат." << endl;
22         return -1;
23     }
24
25     va_list args;
26     va_start(args, numPoints);
27
28     // Збереження координат
29     double* coords = new double[numPoints];
30     for (int i = 0; i < numPoints; ++i) {
31         coords[i] = va_arg(args, double);
32     }
33     va_end(args);
```

```

35 // Знаходимо найдовшу діагональ
36 double maxDiagonal = 0;
37 int p1 = 0, p2 = 0;
38 for (int i = 0; i < numPoints / 2; ++i) {
39     for (int j = i + 1; j < numPoints / 2; ++j) {
40         double length = sideLength(coords[2 * i], coords[2 * i + 1], coords[2 * j], coords[2 * j + 1]);
41         if (length > maxDiagonal) {
42             maxDiagonal = length;
43             p1 = i;
44             p2 = j;
45         }
46     }
47 }
48
49 // Обчислення площі трикутника, що містить найдовшу діагональ
50 double area = 0;
51 for (int i = 0; i < numPoints / 2; ++i) {
52     if (i != p1 && i != p2) {
53         area = square(
54             coords[2 * p1], coords[2 * p1 + 1],
55             coords[2 * p2], coords[2 * p2 + 1],
56             coords[2 * i], coords[2 * i + 1]
57         );
58         break;
59     }
60 }
61
62 delete[] coords;
63 return area;
64 }
65
66 int main() {
67     // Приклад використання
68     double x1 = 0, y1 = 0, x2 = 4, y2 = 0, x3 = 2, y3 = 3;
69     cout << "Довжина сторони: " << sideLength(x1, y1, x2, y2) << endl;
70     cout << "Площа трикутника: " << square(x1, y1, x2, y2, x3, y3) << endl;
71
72     cout << "Площа трикутника з найдовшою діагоналлю: "
73         << square1(8, 0.0, 0.0, 4.0, 0.0, 4.0, 3.0, 0.0, 3.0) << endl;
74
75     return 0;
76 }
77

```

Посилання на pull-request:

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/658/files#diff-f25472d8a00cb76c5463150dc5a50f30b127910f873781a732b0a2ae554833d3](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/658/files#diff-f25472d8a00cb76c5463150dc5a50f30b127910f873781a732b0a2ae554833d3)

Завдання №4

```

1  #include <iostream>
2
3  void removeEvenRows(int**& arr, int& rows, int cols) {
4      int newRowIndex = 0;
5      for (int i = 0; i < rows; ++i) {
6          if (i % 2 == 0) {
7              delete[] arr[i]; // Видаляємо парний рядок
8          } else {
9              arr[newRowIndex++] = arr[i]; // Зміщуємо непарні рядки
10         }
11     }
12     rows = newRowIndex; // Оновлюємо кількість рядків
13 }
14
15 void printArray(int** arr, int rows, int cols) {
16     for (int i = 0; i < rows; ++i) {
17         for (int j = 0; j < cols; ++j) {
18             std::cout << arr[i][j] << " ";
19         }
20         std::cout << std::endl;
21     }
22 }
23
24 int main() {
25     int rows = 4, cols = 3;
26     int** arr = new int*[rows];
27
28     // Заповнюємо масив
29     for (int i = 0; i < rows; ++i) {
30         arr[i] = new int[cols];
31         for (int j = 0; j < cols; ++j) {
32             arr[i][j] = i * cols + j; // Наприклад, послідовність чисел
33         }
34     }
35
36     std::cout << "Before removing even rows:" << std::endl;
37     printArray(arr, rows, cols);
38
39     removeEvenRows(arr, rows, cols);
40
41     std::cout << "After removing even rows:" << std::endl;
42     printArray(arr, rows, cols);
43
44     // Очистка пам'яті
45     for (int i = 0; i < rows; ++i) {
46         delete[] arr[i];
47     }
48     delete[] arr;
49
50     return 0;
51 }
52

```

Посилання на pull-request:

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/658/files#diff-96952c5a24f0ca5396d33c356c3429ac6a148744921828b175db63441132bc36](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/658/files#diff-96952c5a24f0ca5396d33c356c3429ac6a148744921828b175db63441132bc36)

Завдання №5

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  void removeElementsBetweenZeros(vector<int>& arr) {
7      int firstZero = -1; // Індекс першого нуля
8      for (int i = 0; i < arr.size(); ++i) {
9          if (arr[i] == 0) {
10             if (firstZero == -1) {
11                 firstZero = i; // Зберігаємо індекс першого нуля
12             } else {
13                 // Видаляємо елементи між першим і другим нулем
14                 arr.erase(arr.begin() + firstZero + 1, arr.begin() + i);
15                 break; // Перериваємо після видалення
16             }
17         }
18     }
19 }
20
21 void printArray(const vector<int>& arr) {
22     for (int val : arr) {
23         cout << val << " ";
24     }
25     cout << endl;
26 }
27
28 int main() {
29     vector<int> arr = {1, 2, 0, 4, 5, 0, 6, 7};
30
31     cout << "Before removing elements between zeros:" << endl;
32     printArray(arr);
33
34     removeElementsBetweenZeros(arr);
35
36     cout << "After removing elements between zeros:" << endl;
37     printArray(arr);
38
39     return 0;
40 }
41
```

Посилання на pull-request:

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/658/files#diff-2fc8d039f51f0969ff70c6deab1709e599ffcde0b89238072dc8014a6f050581](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/658/files#diff-2fc8d039f51f0969ff70c6deab1709e599ffcde0b89238072dc8014a6f050581)

## Завдання №6

```
1  #include <iostream>
2  #include <vector>
3  #include <string>
4  using namespace std;
5
6  int main() {
7      // Створення масивів для зберігання назв книг і їх доступності
8      vector<string> books = {"C++ Programming", "Data Structures", "Algorithms", "Computer Networks"};
9      vector<bool> availability = {true, true, true, true}; // true - книга доступна, false - книга позичена
10
11      int choice; // Змінна для збереження вибору користувача
12      bool continue_program = true; // Флаг для продовження роботи програми
13
14      while (continue_program) {
15          // Головне меню
16          cout << "\nБібліотека\n";
17          cout << "1. Перерахувати всі книги\n";
18          cout << "2. Позичити книгу\n";
19          cout << "3. Повернути книгу\n";
20          cout << "4. Вийти\n";
21          cout << "Виберіть опцію (1-4): ";
22          cin >> choice;
23
24          switch (choice) {
25              case 1: {
26                  // Перерахування всіх книг
27                  cout << "\nОписок книг:\n";
28                  for (int i = 0; i < books.size(); i++) {
29                      cout << i + 1 << ". " << books[i];
30                      if (availability[i]) {
31                          cout << " - доступна\n";
32                      } else {
33                          cout << " - позичена\n";
34                      }
35                  }
36                  break;
37              }
38              case 2: {
```



```

38 case 2: {
39     // Позичити книгу
40     int book_number;
41     cout << "\nВведіть номер книги, яку хочете позичити (1-" << books.size() << "): ";
42     cin >> book_number;
43
44     if (book_number < 1 || book_number > books.size()) {
45         cout << "Невірний номер книги. Спробуйте ще раз.\n";
46         goto main_menu; // Перехід до головного меню, якщо номер книги некоректний
47     }
48
49     // Перевірка на доступність книги
50     if (availability[book_number - 1]) {
51         availability[book_number - 1] = false;
52         cout << "Ви успішно позичили книгу: " << books[book_number - 1] << endl;
53     } else {
54         cout << "Ця книга вже позичена.\n";
55     }
56     break;
57 }
58 case 3: {
59     // Повернути книгу
60     int book_number;
61     cout << "\nВведіть номер книги, яку хочете повернути (1-" << books.size() << "): ";
62     cin >> book_number;
63
64     if (book_number < 1 || book_number > books.size()) {
65         cout << "Невірний номер книги. Спробуйте ще раз.\n";
66         goto main_menu; // Перехід до головного меню, якщо номер книги некоректний
67     }
68
69     // Повернення книги
70     if (!availability[book_number - 1]) {
71         availability[book_number - 1] = true;
72         cout << "Ви успішно повернули книгу: " << books[book_number - 1] << endl;
73     } else {
74         cout << "Ця книга не була позичена.\n";
75     }
76     break;
77 }
78 case 4:
79     // Вихід з програми
80     cout << "Дякуємо за використання бібліотеки! До побачення.\n";
81     continue_program = false; // Завершуємо програму
82     break;
83 default:
84     cout << "Невірний вибір. Спробуйте ще раз.\n";
85     goto main_menu; // Перехід до головного меню в разі неправильного вибору
86 }
87

```

```

88         // Запит користувача, чи хоче він виконати ще одну операцію
89         char again;
90         cout << "\nБажаєте виконати іншу операцію? (y/n): ";
91         cin >> again;
92
93         if (again == 'n' || again == 'N') {
94             continue_program = false; // Завершуємо програму, якщо користувач вибрав "n"
95         }
96
97     main_menu:
98         continue_program;
99     }
100
101     return 0;
102 }
103

```

Посилання на pull-request:

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/658/files#diff-00d0cd0e9988ebbd84ba5247da463ddef147489777b4b0075c647c8005f4eada](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/658/files#diff-00d0cd0e9988ebbd84ba5247da463ddef147489777b4b0075c647c8005f4eada)

Завдання №7

```

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<pair<int, int>> companies(n);
12      for (int i = 0; i < n; i++) {
13          int length;
14          cin >> length;
15          companies[i] = {length, i + 1}; // {довжина офісу, номер компанії}
16      }
17
18      // Сортуємо компанії за довжиною офісу
19      sort(companies.begin(), companies.end());
20
21      // Виводимо порядок компаній
22      for (int i = 0; i < n; i++) {
23          cout << companies[i].second << " ";
24      }
25
26      return 0;
27  }
28

```

Посилання на pull-request:

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/658/files#diff-544febbb5598314f4769d3b63c1b328bae018979328bc1a4f524982aae25a559](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/658/files#diff-544febbb5598314f4769d3b63c1b328bae018979328bc1a4f524982aae25a559)

## 5. Результати виконання завдань, тестування та фактично затрачений час:

Завдання №1

```
● Сума 7 членів ряду: 14.3548
```

Час затрачений на виконання завдання: 30 хв

Завдання №2

```
X = 0.1 SN = inf SE = 0.100167 Y = 0.100167
X = 0.19 SN = inf SE = 0.191145 Y = 0.191145
X = 0.28 SN = inf SE = 0.283673 Y = 0.283673
X = 0.37 SN = inf SE = 0.3785 Y = 0.3785
X = 0.46 SN = inf SE = 0.476395 Y = 0.476395
X = 0.55 SN = inf SE = 0.578152 Y = 0.578152
X = 0.64 SN = inf SE = 0.684594 Y = 0.684594
X = 0.73 SN = inf SE = 0.796586 Y = 0.796586
X = 0.82 SN = inf SE = 0.915034 Y = 0.915034
X = 0.91 SN = inf SE = 1.0409 Y = 1.0409
X = 1 SN = inf SE = 1.1752 Y = 1.1752
```

Час затрачений на виконання завдання: 1 год

Завдання №3

```
Довжина сторони: 4
Площа трикутника: 6
Площа трикутника з найдовшою діагоналлю: 6
```

Час затрачений на виконання завдання: 1 год

Завдання №4

```
Before removing even rows:
```

```
0 1 2
```

```
3 4 5
```

```
6 7 8
```

```
9 10 11
```

```
After removing even rows:
```

```
3 4 5
```

```
9 10 11
```

Час затратений на виконання завдання: 1 год

Завдання №5

```
Before removing elements between zeros:
```

```
1 2 0 4 5 0 6 7
```

```
After removing elements between zeros:
```

```
1 2 0 0 6 7
```

Час затратений на виконання завдання: 40 хв

Завдання №6

```

Бібліотека
1. Перерахувати всі книги
2. Позичити книгу
3. Повернути книгу
4. Вийти
Виберіть опцію (1-4): 1

Список книг:
1. C++ Programming - доступна
2. Data Structures - доступна
3. Algorithms - доступна
4. Computer Networks - доступна

Бажаєте виконати іншу операцію? (y/n): y

Бібліотека
1. Перерахувати всі книги
2. Позичити книгу
3. Повернути книгу
4. Вийти
Виберіть опцію (1-4): 2

Введіть номер книги, яку хочете позичити (1-4): 2
Ви успішно позичили книгу: Data Structures

Бажаєте виконати іншу операцію? (y/n): y

Бібліотека
1. Перерахувати всі книги
2. Позичити книгу
3. Повернути книгу
4. Вийти
Виберіть опцію (1-4): 1

Список книг:
1. C++ Programming - доступна
2. Data Structures - позичена
3. Algorithms - доступна
4. Computer Networks - доступна

Бажаєте виконати іншу операцію? (y/n): y

Бібліотека
1. Перерахувати всі книги
2. Позичити книгу
3. Повернути книгу
4. Вийти
Виберіть опцію (1-4): 4
Дякуємо за використання бібліотеки! До побачення.

Бажаєте виконати іншу операцію? (y/n): n

```

Час затрачений на виконання завдання: 2 год

## Завдання №7

```

7
2 6 8 9 1 3 5
5 1 6 7 2 3 4

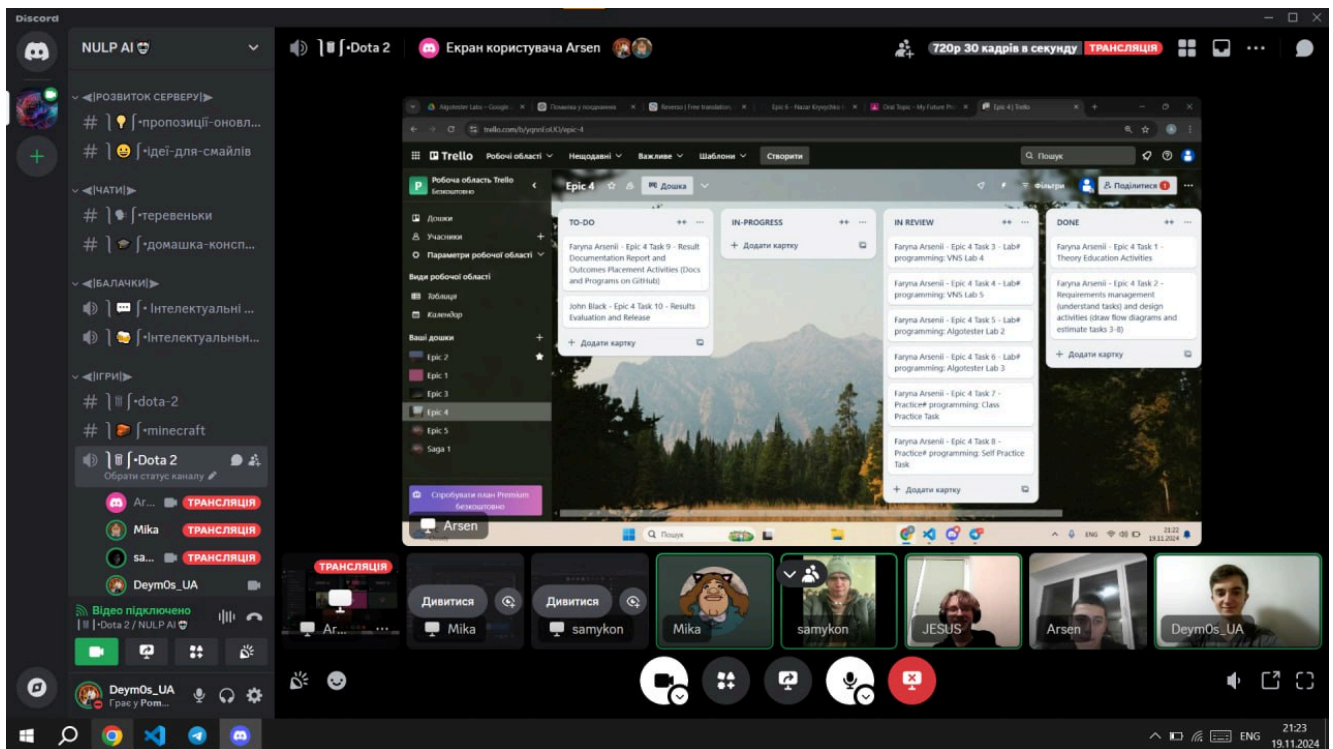
```

Час затрачений на виконання завдання: 30 хв

Час затрачений на виконання завдання: 30 хв

## 6. Кооперація з командою:

- Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло



## Висновки:

Протягом виконання завдань я закріпив основні поняття роботи з масивами, структурами, динамічним виділенням пам'яті та рекурсією. Я реалізував різноманітні алгоритми для пошуку, сортування, обробки даних і оптимізації. Окремо приділив увагу перевантаженню функцій, передачі масивів у функції та використанню вкладених структур. У задачах застосовувались статичні та динамічні масиви, що дозволило краще зрозуміти їх відмінності та переваги. У результаті я отримав практичні навички роботи з базовими алгоритмами та структурою програм у C++, що допоможе в майбутньому вирішувати складніші задачі.