Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 3**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: «Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія.»

***Виконав:***

студент групи ШІ-11

Іванов Олексій Олександрович

# **Тема роботи:**

Побудова лінійних, рекурсивних та розгалужених алгоритмів для розв’язання задач

# **Мета роботи:**

Ознайомитися з поняттями лінійного та розгалуженого алгоритмів, навчитись будувати лінійні та розгалужені алгоритми для розв’язання задач, ознайомитись зі статичним поліморфізмом

# **Виконання роботи:**

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

Завдання № 1 VNS Lab 2 - Task 1-1

* Варіант завдання - 24
* Деталі завдання

1. Необхідно обчислити суму ряду із заданою точністю або n-ну кількість перших елементів (залежно від варіанту)
2. Необхідно обчислювати суму за допомогою рекурентної формули an+1 = an\*x, де x - відношення послідовних членів

* Важливі деталі для врахування в імплементації програми

1. Існування вбудованої функції e^n
2. Існування вбудованої функції sqrt (square root) та pow (power - для 1/ n працює як корінь n-го степеня)

Завдання № 2 VNS Lab 3 - Task 1-1

* Варіант завдання - 24
* Деталі завдання

Обчислити f(x), використовуючи її розклад в степеневий ряд у двох випадках:

а) для заданого n;

б) для заданої точності ε (ε=0.0001).

Для порівняння знайти точне значення функції.

* Важливі деталі для врахування в імплементації програми

1. Використати тип double
2. Використати ряд Маклорена для заданої функції
3. Вивести рекурентну формулу наступного члена ряду

Завдання № 3 VNS Lab 7 - Task 1-1

* Варіант завдання - 24
* Деталі завдання

Написати функцію/макрос, що знаходить довжину сторони за координатами його точок, написати функцію зі змінною кількістю параметрів для знаходження площі опуклого n-кутника

* Важливі деталі для врахування в імплементації програми

1. Написати програму в C-стилі
2. Використати змінну кількість параметрів для функції площі

Завдання № 4 VNS Lab 7 - Task 1-2

* Варіант завдання - 24
* Деталі завдання

Написати перевантажені функції й основну програму, що їх викликає:

а) для масиву цілих чисел знищує всі парні елементи з масиву

б) для рядка знищує всі парні слова

* Важливі деталі для врахування в імплементації програми

Необхідно використати перевантажені процедури та викликати їх з main

Завдання № 5 Class Practice Work

* Деталі завдання

Програма керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

* Важливі деталі для врахування в імплементації програми

Програма повинна вміти:

1. Перерахувати всі книги.
2. Дозволити взяти книгу (за наявності).
3. Дозволити повернення книги.

Структури даних

1. Використовуйте масив або вектор для зберігання назв книг.
2. Використовуйте інший масив або вектор для збереження стану доступності кожної книги.

Завдання № 6 Self Practice Work (Algotester Lab 2)

* Варіант завдання - 3
* Деталі завдання

Вам дано масив цілих чисел розміром N, на першій та останній клітинці розміщено по дрону. Вони одночасно злітають і рухаються назустріч один одному.

* Важливі деталі для врахування в імплементації програми

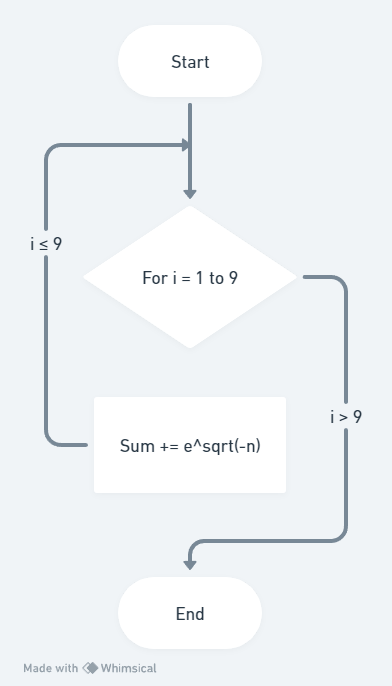
Вони це роблять доти:

1. доки 2 дрони не опиняться в одній клітинці - Collision;
2. доки лівий дрон не опиниться справа від правого - Miss;
3. доки вони не зупиняться один навпроти одного - Stopped

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

Програма № 1 “Використання основних операторів мови С”

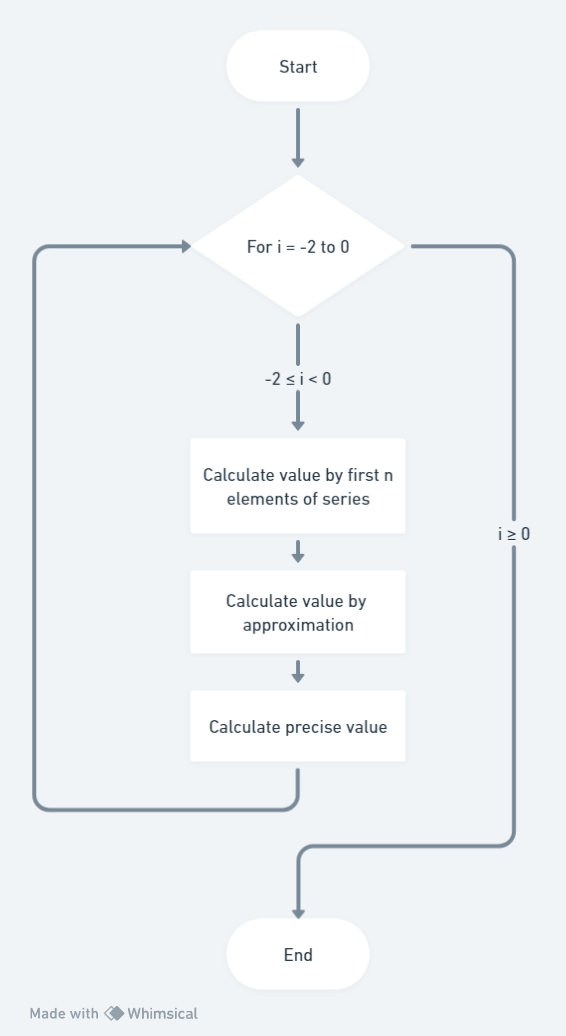
* Блок-схема

Рисунок 1: Блок-схема до програми № 1

* Планований час на реалізацію — 20 хвилин

Програма № 2 "Обчислення функцій з використанням їхнього розкладу в степеневий ряд"

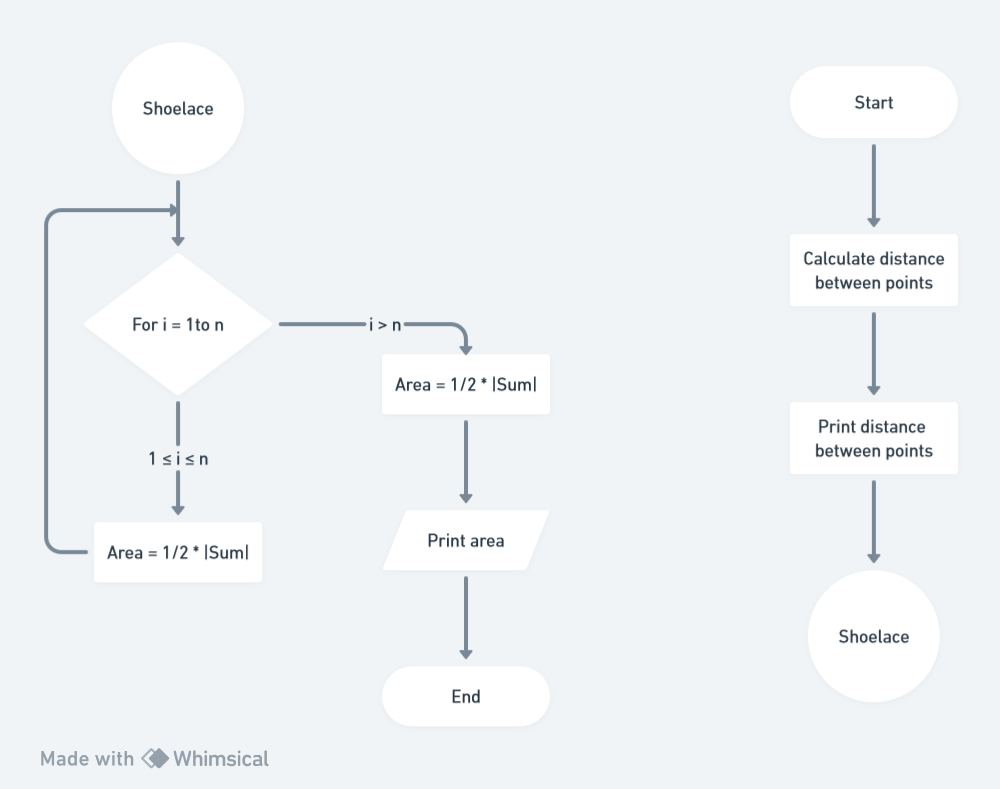
* Блок-схема

Рисунок 2: Блок-схема до програми № 2

- Планований час на реалізацію — 1 година

Програма № 3 "Функції зі змінною кількістю параметрів в С++"

* Блок-схема

Рисунок 3: Блок-схема до програми № 3

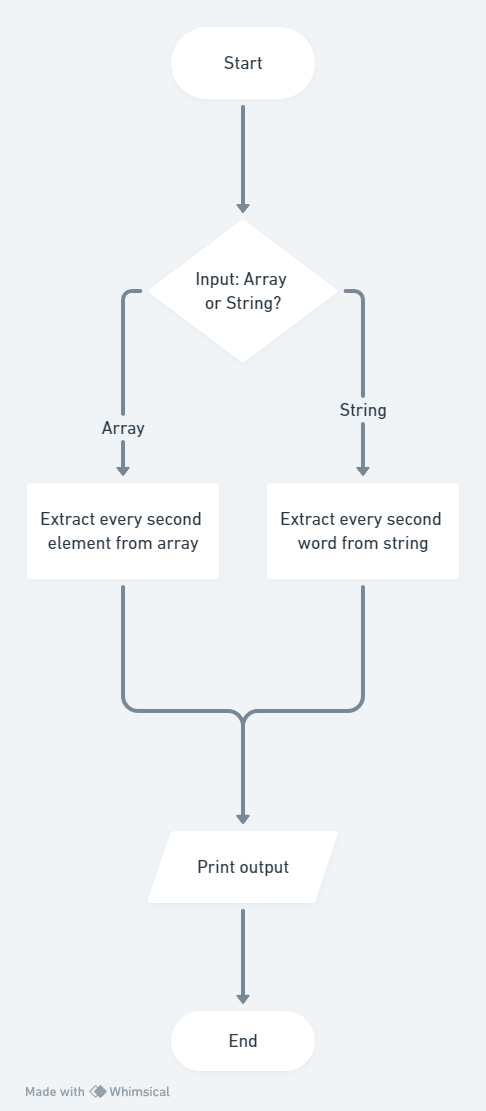
* Планований час на реалізацію — 20 хвилин
* Важливі деталі під час реалізації:

Формула для обчислення площі опуклого багатокутника:

Рисунок 4: Формула площі Гауса

Програма № 4 "Перевантаження функцій в С++"

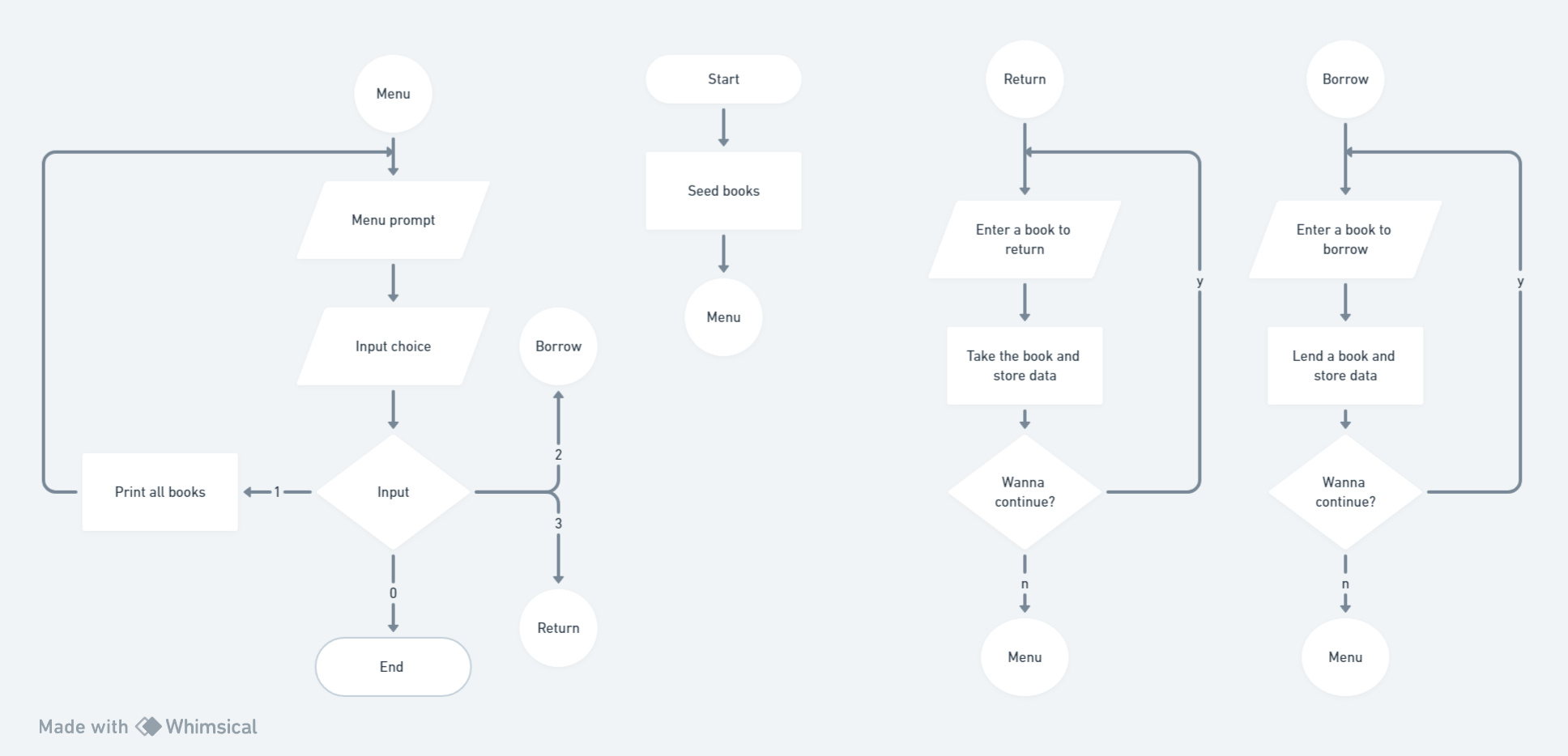
* Блок-схема

Рисунок 5: Блок-схема до програми № 4

* Планований час на реалізацію — 40 хвилин

Програма № 5 Бібліотека

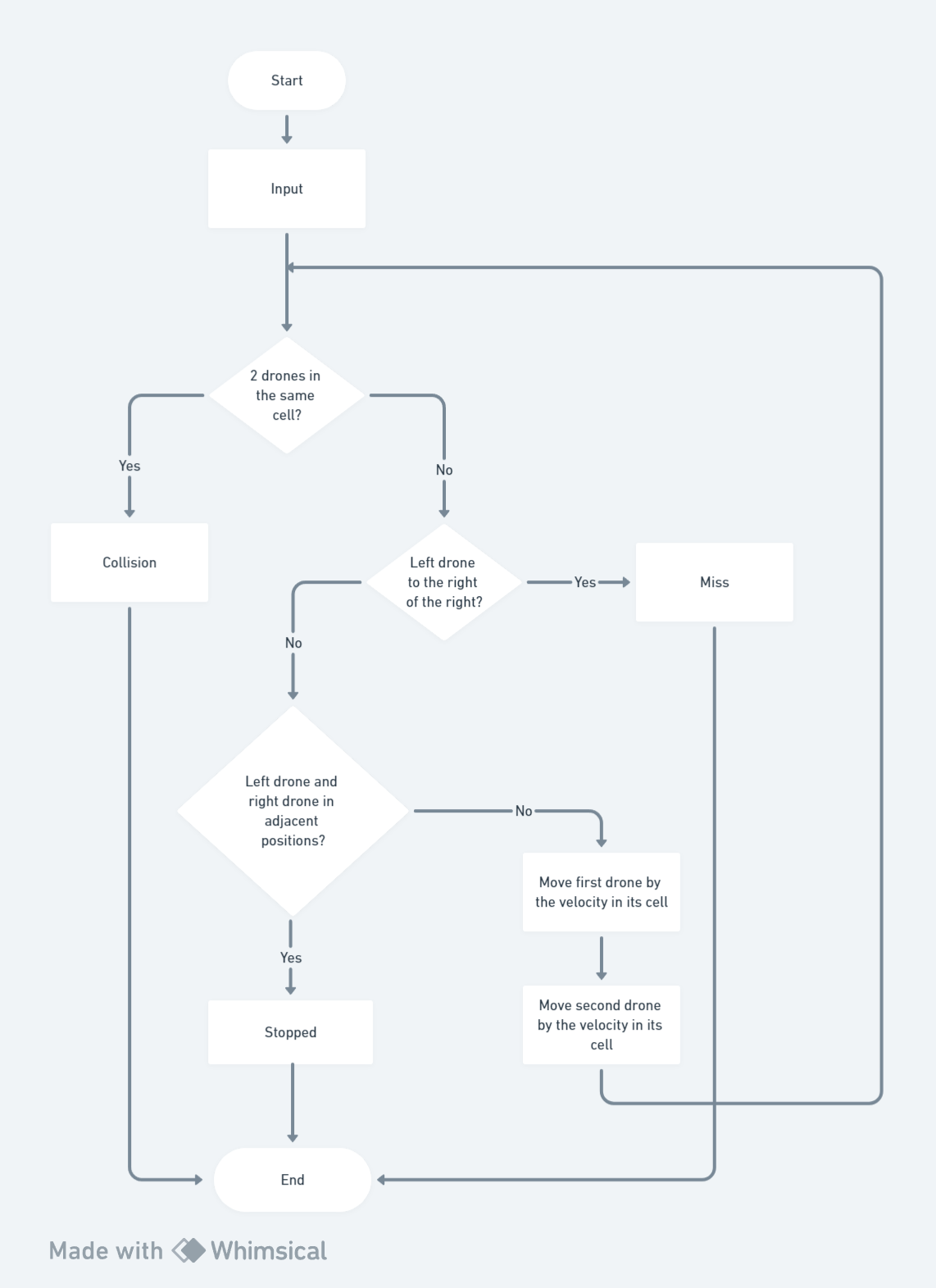
* Блок-схема

Рисунок 6: Блок-схема до програми № 5

* Планований час на реалізацію — 40 хвилин

Програма № 6 Algotester Lab 2

* Блок-схема

Рисунок 7: Блок-схема до програми № 6

* Планований час на реалізацію — 20 хвилин

## **3. Код програм з посиланням на зовнішні ресурси:**

Завдання № 1 Сума перших членів ряду

#include <iostream>  
#include <cmath>  
using std::cout, std::endl;  
using SeriesElement = double(unsigned int);  
  
double evalSeriesElement(unsigned int n) {  
 if (n == 0) throw;  
 return exp(-sqrt(n));  
}  
  
double evalSumOfSeries(SeriesElement element, unsigned int n) {  
 if (n == 0) return 0;  
  
 double sum = 0;  
 for (int i = 1; i <= n; ++i)  
 sum += element(i);  
  
 return sum;  
}  
  
static const int count = 9;  
  
int main() {  
 cout << evalSumOfSeries(evalSeriesElement, count);  
 return 0;  
}

Код програми № 1 ([посилання на файл в пул-запиті](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/394/files" \l "diff-e981f5889fbf0b0f6e465aaf8afb77bb5254ace65b351ac2af4ad82248231936))

Завдання № 2 Ряд Маклорена для обрахунку функції

#include <iostream>  
#include <iomanip>  
#include <cmath>  
  
using std::cout, std::setprecision, std::fixed;  
  
/\*  
 \* 1  
 \* y = ln(------------)  
 \* 2+2x+x^2  
 \*/  
  
double yByStdMethod(double x) {  
 return log(1 / (2 + 2 \* x + pow(x, 2)));  
}  
  
double yByPartialSum(double x, double n) {  
 // first element -(x+1)^2  
 double current = -pow(x + 1, 2);  
 double prev, sum = current;  
  
 for (int i = 1; i <= n; ++i) {  
 prev = current;  
 // recurrence relation  
 current = prev \* (-n \* pow(x + 1, 2)) / (n + 1);  
 sum += current;  
 }  
  
 return sum;  
}  
  
double yByEpsilon(double x, double epsilon = 0.0001) {  
 // first element -(x+1)^2  
 double current = -pow(x + 1, 2);  
 double prev = 0, sum = current;  
  
 double diff = fabs(current - prev);  
 for (int n = 1; diff >= epsilon; ++n) {  
 diff = fabs(current - prev);  
 prev = current;  
  
 // recurrence relation  
 current = prev \* (-(n \* pow(x + 1, 2)) / (n + 1));  
 sum += current;  
 }  
  
 return sum;  
}  
  
int main() {  
 double e = 1e-4,  
 n = 40;  
  
 double a = -2, b = -0.1, k = 10;  
  
 cout << setprecision(10) << fixed;  
  
 double x = a;  
 while (x <= b) {  
 cout << "X = " << x << "\t"  
 << "SN = " << yByPartialSum(x, n) << "\t\t"  
 << "SE = " << yByEpsilon(x, e) << "\t\t"  
 << "Y = " << yByStdMethod(x) << "\n";  
  
 x += (b - a) / k;  
 }  
}

Код програми № 2 ([посилання на файл в пул-запиті](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/394/files" \l "diff-a144eaff7e46cb99c246709dbdf9e803f5fe202ca1cfdb4b4e5b229d2ead7b19))

Завдання № 3 Площа опуклого багатокутника через функцією зі змінною кількістю параметрів

#include <stdio.h>  
#include <stdarg.h>  
#include <math.h>  
  
#define DISTANCE(x1, y1, x2, y2) \  
 sqrt((x2 - x1) \* (x2 - x1) + (y2 - y1) \* (y2 - y1))  
  
typedef struct {  
 double x, y;  
} Point;  
  
Point create\_point(double x, double y) {  
 Point p;  
 p.x = x;  
 p.y = y;  
 return p;  
}  
  
double square\_n(int n, ...);  
double square(Point p1, Point p2, Point p3) {  
 return square\_n(3, p1, p2, p3);  
}  
  
int main() {  
 // Suspected output: sqrt(5) ~ 2.236068  
 printf("Distance between (2, 4) and (3, 2): %lf\n", DISTANCE(2, 4, 3, 2));  
  
 // Suspected output: 2  
 printf("Square area of triangle: %lf\n",  
 square(create\_point(0, 0), create\_point(0, 2), create\_point(2, 0)));  
  
 // Suspected output: 339865  
 printf("Square area of %d-gon: %lf\n", 4, square\_n(  
 4,  
 create\_point(-1000, 500),  
 create\_point(-500, 1000),  
 create\_point(2, 10),  
 create\_point(35, 60)  
 ));  
  
 // Suspected output: 124562.5  
 printf("Square area of %d-gon: %lf\n", 10, square\_n(  
 10,  
 create\_point(51, -20),  
 create\_point(15, 3),  
 create\_point(45, 200),  
 create\_point(100, -100),  
 create\_point(201, 55),  
 create\_point(70, -80),  
 create\_point(25, 333),  
 create\_point(999, 0),  
 create\_point(500, 77),  
 create\_point(5, -6)  
 ));  
  
 // Suspected output: 1446  
 printf("Square area of %d-gon: %lf\n", 5, square\_n(  
 5,  
 create\_point(13, -92),  
 create\_point(44, 0),  
 create\_point(-800, 30),  
 create\_point(27, 2),  
 create\_point(1, 2)  
 ));  
  
 return 0;  
}  
  
double square\_n(int n, ...) {  
 va\_list args;  
 va\_start(args, n);  
  
 Point start = va\_arg(args, Point), // x0, y0  
 current = start, // xi, yi  
 next; // xi+1, yi+1  
  
 double sum = 0;  
 for (int i = 0; i < n - 1; ++i) {  
 next = va\_arg(args, Point);  
 // Calculate the area of a triangle using Shoelace formula  
 sum += current.x \* next.y - next.x \* current.y;  
 current = next;  
 }  
 sum += current.x \* start.y - start.x \* current.y;  
  
 va\_end(args);  
  
 double result = fabs(sum) / 2;  
 return result;  
}

Код програми № 3 ([посилання на файл в пул-запиті](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/394/files" \l "diff-d3b86392294ea9438846c4594f404588667cb4e817a298b2b05df89b18fa2096))

Завдання № 4 Очистка від елементів з парним індексом

#include <iostream>  
#include <sstream>  
  
using namespace std;  
  
int\* cleanse(const int\* arr, int n) {  
 int\* result = new int[n / 2 + n % 2];  
 int j = 0;  
 for (int i = 0; i < n; ++i) {  
 if (i % 2 == 0) {  
 result[j] = arr[i];  
 ++j;  
 }  
 }  
 return result;  
}  
  
string cleanse(const string& str, char separator = ' ') {  
 string result, buf;  
 stringstream ss(str);  
  
 int n = 0;  
 while (getline(ss, buf, separator)) {  
 if (n % 2 == 0) result += buf + separator;  
 n++;  
 }  
  
 return result;  
}  
  
int main() {  
 int n;  
 cout << "Enter count of elements in array: ";  
 cin >> n;  
  
 int\* arr = new int[n];  
 cout << "Enter elements of array: ";  
 for (int i = 0; i < n; ++i) cin >> arr[i];  
  
 int new\_size = n / 2 + n % 2;  
 int\* result = cleanse(arr, n);  
 cout << "Result: ";  
 for (int i = 0; i < new\_size; ++i) cout << result[i] << ' ';  
  
 cout << endl;  
  
 string str;  
 cout << "Enter string: ";  
 cin.ignore(INT\_MAX, '\n');  
 getline(cin, str);  
  
 cout << "Result: " << cleanse(str) << endl;  
  
 return 0;  
}

Код програми № 4 ([посилання на файл в пул-запиті](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/394/files" \l "diff-6454bb42188b2c84f14eff8862a8559d93dbe3c45bb23b0c5c0e32784c5e3dbb))

Завдання № 5 Бібліотека

#include "../books/book.h"  
#include <iostream>  
#include <string>  
#include <vector>  
using namespace std;  
  
string menu\_prompt = "Menu\n"  
 "1 - Enumerate books\n"  
 "2 - Borrow a book\n"  
 "3 - Return a borrowed book\n"  
 "0 - Exit\n";  
  
string enter\_choice = "Enter your choice: ",  
 wrong\_input = "Unexpected input",  
 continuation\_prompt = "Do you wanna continue? (Y/n): ",  
 selection\_prompt = "Select a book: ";  
  
void seed\_books(vector<Book>& \_books);  
void enumerate\_books(const vector<Book>& books);  
  
int main() {  
 vector<Book> books;  
 seed\_books(books);  
  
 string input\_buf;  
  
 char action\_choice;  
 char continuation\_choice;  
  
 bool should\_continue;  
 bool exit = false;  
  
 while (!exit) {  
 goto menu\_start;  
  
 wrong\_input:  
 cout << wrong\_input << endl;  
 goto menu\_start;  
  
 menu\_start:  
 should\_continue = true;  
 cout << menu\_prompt << enter\_choice;  
 getline(cin, input\_buf);  
  
 // considering only first character of the input  
 if (input\_buf.size() > 1) goto wrong\_input;  
// else if (input\_buf.empty()) goto menu\_start;  
 action\_choice = input\_buf[0];  
  
 do {  
 switch (action\_choice) {  
 case '1':  
 enumerate\_books(books);  
 break;  
 case '2': {  
 cout << selection\_prompt;  
 enumerate\_books(books);  
 cout << enter\_choice;  
  
 size\_t book\_index;  
 cin >> book\_index;  
 cin.ignore(INT\_MAX, '\n');  
  
 // shifting from user input (1, ...) to vector index  
 book\_index--;  
  
 if (book\_index >= books.size()) goto wrong\_input;  
  
 Book &selected\_book = books[book\_index];  
  
 if (selected\_book.getIsAvailable()) {  
 selected\_book.setIsAvailable(false);  
 cout << "Successfully borrowed \""  
 << selected\_book.getName()  
 << "\" book" << endl;  
 } else  
 cout << "The book \"" << selected\_book.getName()  
 << "\" is not available." << endl;  
  
 break;  
 }  
 case '3': {  
 cout << selection\_prompt;  
 enumerate\_books(books);  
 cout << enter\_choice;  
  
 size\_t book\_index;  
 cin >> book\_index;  
 cin.ignore(INT\_MAX, '\n');  
  
 // shifting from user input (1, ...) to vector index  
 book\_index--;  
  
 if (book\_index >= books.size()) goto wrong\_input;  
  
 Book &selected\_book = books[book\_index];  
  
 if (!selected\_book.getIsAvailable()) {  
 selected\_book.setIsAvailable(true);  
 cout << "Successfully returned \""  
 << selected\_book.getName()  
 << "\" book" << endl;  
 } else  
 cout << "The book \"" << selected\_book.getName()  
 << "\" is already in library." << endl;  
  
 break;  
 }  
 case '0':  
 exit = true;  
 should\_continue = false;  
 break;  
 default:  
 goto wrong\_input;  
 }  
  
 if (!should\_continue || action\_choice == '1') break;  
  
 cout << continuation\_prompt;  
 getline(cin, input\_buf);  
 // considering only first character of the input  
 if (input\_buf.size() != 1) goto wrong\_input;  
 continuation\_choice = (char) tolower(input\_buf[0]);  
  
 if (continuation\_choice == 'y')  
 should\_continue = true;  
 else if (continuation\_choice == 'n')  
 should\_continue = false;  
 else goto wrong\_input;  
 } while (should\_continue);  
 }  
  
 cout << "Bye";  
 return 0;  
}  
  
void seed\_books(vector<Book>& \_books) {  
 \_books.emplace\_back("1984", true);  
 \_books.emplace\_back("Fahrenheit 451", true);  
 \_books.emplace\_back("Harry Potter", true);  
 \_books.emplace\_back("The Hobbit", true);  
 \_books.emplace\_back("The Witcher", true);  
 \_books.emplace\_back("The Hitchhiker's Guide to the Galaxy", true);  
 \_books.emplace\_back("The Restaurant at the End of the Universe", true);  
 \_books.emplace\_back("Life, the Universe and Everything", true);  
 \_books.emplace\_back("So Long, and Thanks for All the Fish", true);  
 \_books.emplace\_back("Mostly Harmless", true);  
 \_books.emplace\_back("The Metamorphosis", true);  
}  
  
void enumerate\_books(const vector<Book>& books) {  
 for (int i = 0; i < books.size(); ++i)  
 cout << i + 1 << " - " << books[i].toString() << endl;  
}

Код програми № 5 ([посилання на](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/394/files" \l "diff-2b81edcd616c398d5b09566b2d36d2403c067d0b01d66948213a465022db6059) main.cpp в пул-запиті)

#ifndef book\_h  
#define book\_h  
  
#include <string>  
#include <utility>  
using std::string;  
  
class Book {  
private:  
 string name;  
 bool isAvailable;  
public:  
 explicit Book(string name = "", bool isAvailable = true) {  
 this->name = std::move(name);  
 this->isAvailable = isAvailable;  
 }  
  
 string getName() const;  
 void setName(string Name);  
  
 bool getIsAvailable() const;  
 void setIsAvailable(bool IsAvailable);  
  
 string toString() const;  
};  
  
#endif //book\_h

Код програми № 5 ([посилання на](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/394/files" \l "diff-5243be37c5d38c0f4f360169d823454fa3c865724d2dd58018813f4cd8c08de2) book.h в пул-запиті)

#include <utility>  
#include "book.h"  
  
string Book::getName() const {  
 return this->name;  
}  
void Book::setName(string Name) {  
 this->name = std::move(Name);  
}  
  
bool Book::getIsAvailable() const {  
 return this->isAvailable;  
}  
void Book::setIsAvailable(bool IsAvailable) {  
 this->isAvailable = IsAvailable;  
}  
  
string Book::toString() const {  
 return this->name + " (" +  
 (isAvailable ? "Available" : "Unavailable")  
 + ")";  
}

Код програми № 5 ([посилання на](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/394/files" \l "diff-e87f5011d332d7e580714153a6a41fd052c10e9faadbdb336b79a803483ca4eb) book.cpp в пул-запиті)

Завдання № 6 Дрони

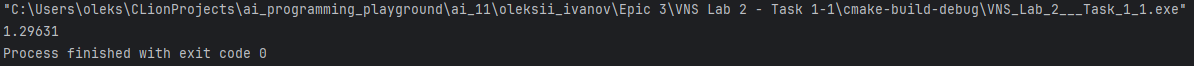
#include <iostream>  
using namespace std;  
  
int main() {  
 int n;  
 cin >> n;  
  
 int\* speeds = new int[n];  
 for (int i = 0; i < n; i++) {  
 cin >> speeds[i];  
 }  
  
 // we have one drone on both sides - start and end  
 int drone\_a\_pos = 1, drone\_b\_pos = n;  
  
 string status = "Ok";  
 while (status == "Ok") {  
 if (drone\_a\_pos == drone\_b\_pos)  
 status = "Collision";  
 else if (drone\_a\_pos + 1 == drone\_b\_pos)  
 status = "Stopped";  
 else if (drone\_a\_pos > drone\_b\_pos)  
 status = "Miss";  
 else {  
 drone\_a\_pos += speeds[drone\_a\_pos - 1];  
 drone\_b\_pos -= speeds[drone\_b\_pos - 1];  
 }  
 }  
  
 cout << drone\_a\_pos << " " << drone\_b\_pos << endl  
 << status;  
  
 delete[] speeds;  
 return 0;  
}

Код програми № 6 ([посилання на файл в пул-запиті](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/394/files" \l "diff-3590eb27e22299bf58717a749ef0aa7eb87904eff54dca3a3ebcd3e25aa899fe))

[**Посилання на pull-request**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/394)

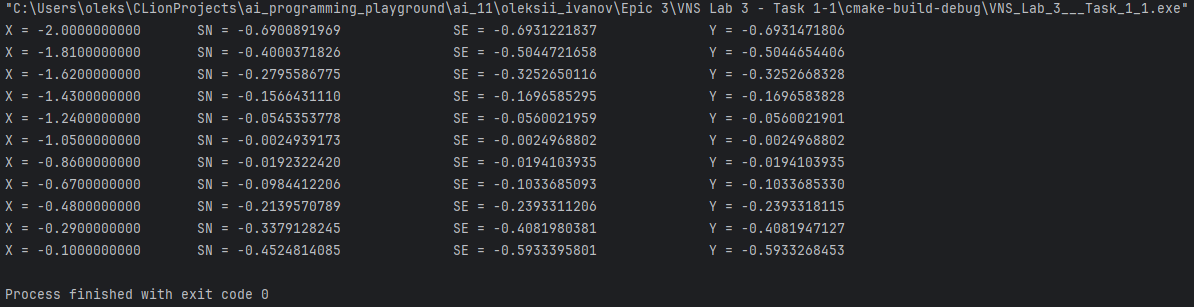
## **4. Результати виконання завдань, тестування та фактично витрачений час:**

Завдання № 1 (Фактично витрачений час — 20 хвилин)

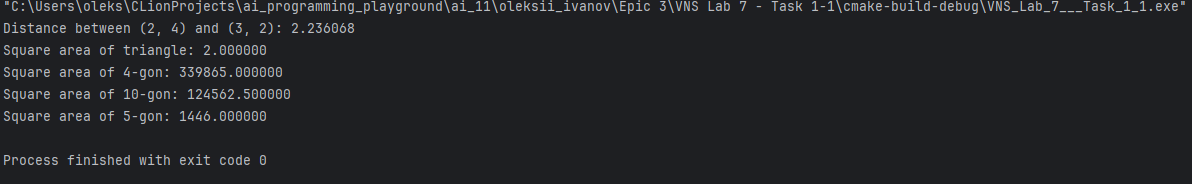
Рисунок 8: Результати виконання програми № 1

Завдання № 2 (Фактично витрачений час — 1 година)

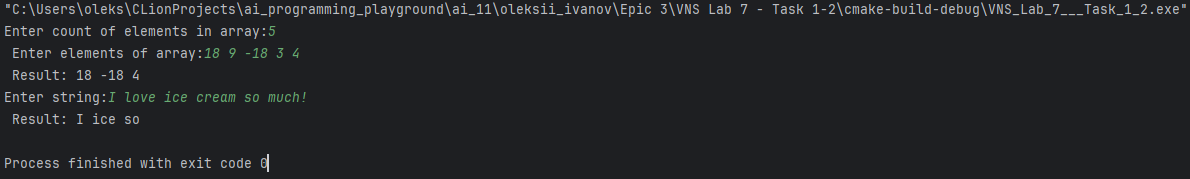
Завдання № 3 (Фактично витрачений час — 20 хвилин)

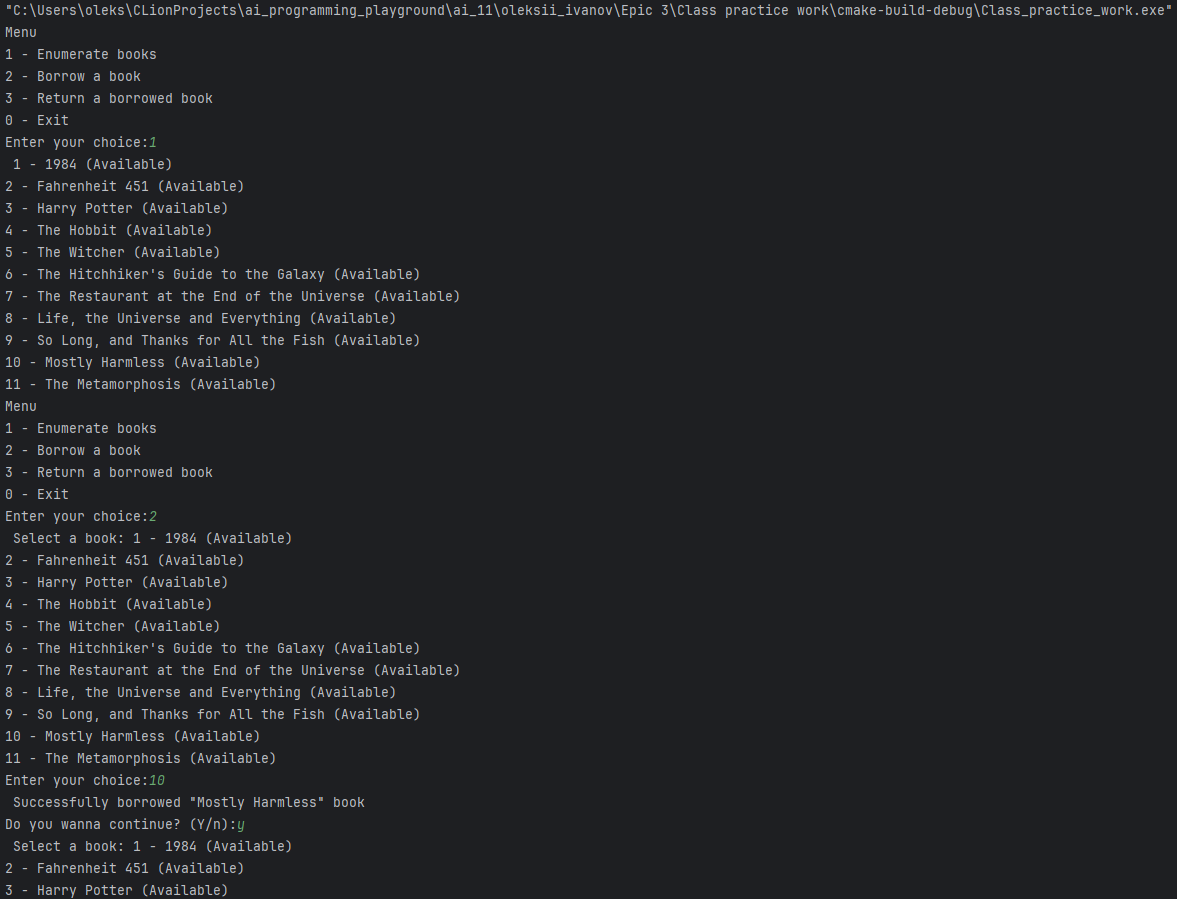
Рисунок 9: Результати виконання програми № 2

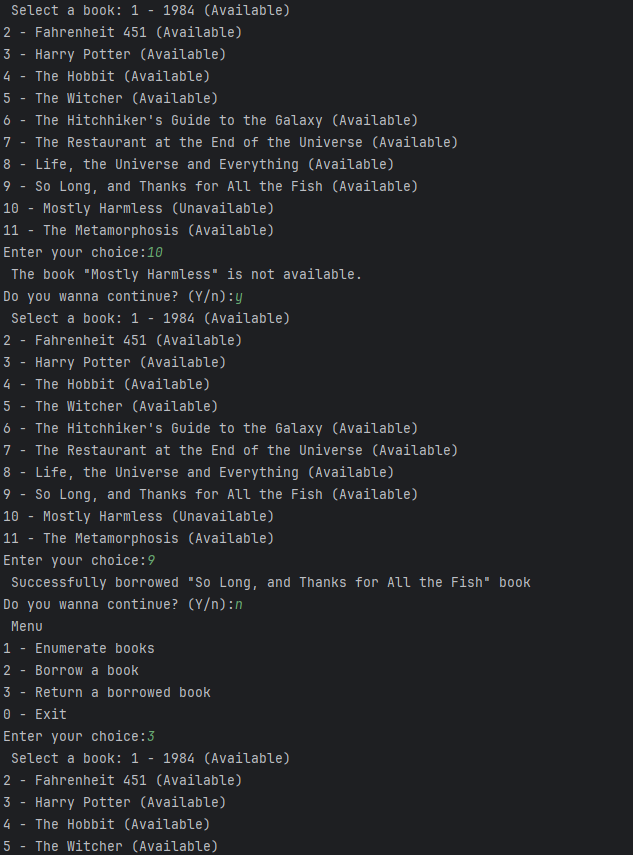
Завдання № 4 (Фактично витрачений час — 50 хвилин)

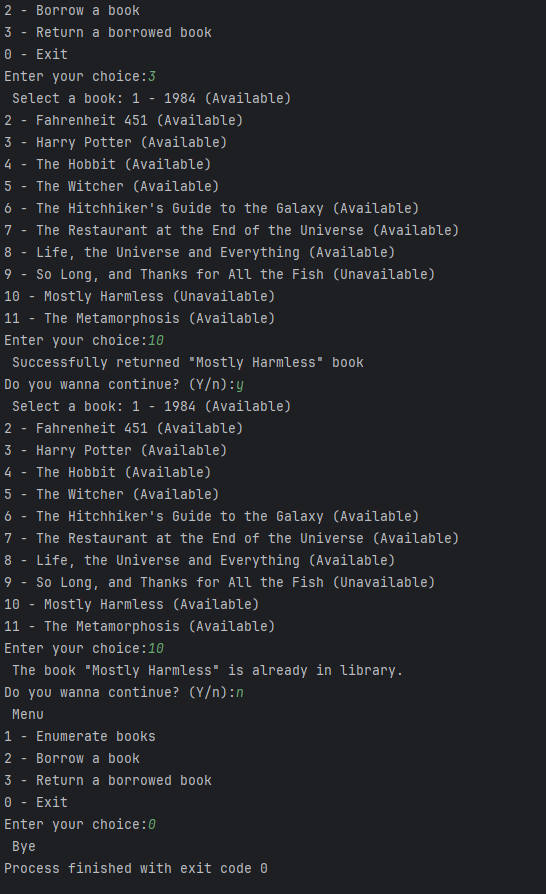
Рисунок 10: Результати виконання програми № 3

Завдання № 5 (Фактично витрачений час — 40 хвилин)

Рисунок 11: Результати виконання програми № 4

Рисунок 12: Результати виконання програми № 5

Рисунок 13: Результати виконання програми № 5

Рисунок 14: Результати виконання програми № 5

Завдання № 6 (Фактично витрачений час — 20 хвилин)

Рисунок 15: Результати виконання програми № 6

# **Висновки:**

У ході виконання роботи були досягнуті такі результати:

* Отримав уявлення про поняття лінійного та розгалуженого алгоритмів.
* Навчився будувати лінійні та розгалужені алгоритми для розв’язання задач.
* Отримав уявлення про статичний поліморфізм.

У рамках роботи було виконано наступні завдання:

* Обчислено суму ряду із заданою точністю або n-ну кількість перших елементів за допомогою рекурентної формули.
* Обчислено f(x), використовуючи її розклад в степеневий ряд у двох випадках: для заданого n і для заданої точності.
* Написано функцію/макрос, що знаходить довжину сторони за координатами його точок, а також функцію зі змінною кількістю параметрів для знаходження площі опуклого n-кутника.
* Написано перевантажені функції, що знищують всі парні елементи з масиву цілих чисел і всі парні слова з рядка.
* Написано програму керування бібліотекою, яка дозволяє перераховувати всі книги, брати книги (за наявності) і повертати книги.
* Написано програму, що моделює рух двох дронів, які одночасно злітають і рухаються назустріч один одному.

У процесі виконання роботи були використані наступні методи та засоби:

* Методи аналізу та синтезу алгоритмів.
* Засоби програмування C/C++.

У результаті виконання роботи було отримано наступні практичні навички:

* Навички написання лінійних та розгалужених алгоритмів.
* Навички використання статичного поліморфізму.