

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



Звіт
про виконання лабораторних та практичних робіт блоку
№ 3
з дисципліни: «Основи програмування»

Виконав:

Студент групи ШІ-11

Лопатін Володимир Дмитрович

Львів 2024

Тема:

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

Мета роботи:

Ознайомитися з принципами роботи циклів і вкладених циклів, а також методами їх завершення. Навчитися використовувати функції, розуміти особливості простору імен, перевантаження функцій та функцій із змінною кількістю параметрів (еліпсис). Дослідити рекурсію як метод побудови алгоритмів і розглянути можливості використання вбудованих функцій. Отримати практичні навички створення ефективних і зрозумілих програм із використанням перелічених елементів програмування.

Теоретичні відомості:

1)Список:

- Цикли та вкладені цикли
- Функції
- Простори імен
- Перевантаження функцій
- Функції зі змінною кількістю параметрів
- Рекурсія
- Вбудовані функції
- Конструкції виходу з циклів

2) Цикли та вкладені цикли:

Був знайомий до того доознайомився під час роботи.
Витрачено 30 хв.

Функції:

Знайомий.
Витратив 10 хвилин.

Простори імен:

Розумів концепцію до того, дорозібрався на практиці
Витратив 15 хвилин.

Перевантаження функцій:

На парі усе зрозумів.
Витратив 20 хвилин.

Функції зі змінною кількістю параметрів:

Пояснили ChatGPT та викладач на парі.
На повне ознайомлення загалом витратив пів години.

Вбудовані функції:

Пояснили на парі, та потім інтуїтивно розібрався в ході виконання завдань.
Витратив 30 хвилин.

Конструкції виходу з циклів:

Пояснювали на парі, потім ще ChatGPT.
Витрачено 30 хвилин.

Виконання роботи:

- 1) Опрацювання завдання та вимог до програм та серидовища:

Завдання №1

«Менеджмент бібліотеки»

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

Вимоги:

- while: продовжувати працювати, доки користувач не вирішить вийти.
- do while: Після кожної операції (позичити, повернути, перерахувати) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.
- for: список усіх книг за допомогою циклу.
- for each: перевірити наявність кожної книги.
- goto: якщо користувач вводить неправильний вибір, використовуйте goto, щоб перенаправити його до головного меню.

Завдання №2

«Практика з алготестера»

Потрібно було знайти відстань, на яку видно дим у залежності від заданих параметрів координат та кількостей пожеж, які побачили та загалом.

Завдання №3

«Лабораторна 2 з ВНС»

Потрібно було знайти суму ряду з наближенням до якогось числа.

Завдання №4

«Лабораторна 3 з ВНС»

Обчислити наближене значення функції залежно від аргументу за розкладанням на ряд Маклорена (з наближенням до якогось числа та в залежності від n).

Завдання №5

«Лабораторна 7 завдання 1 з ВНС»

Завдання стояло створити еліпсис, щоб перевірити чи точки із заданими координатами належать колу з певним радіусом.

Завдання №6

«Лабораторна 7 завдання 2 з ВНС»

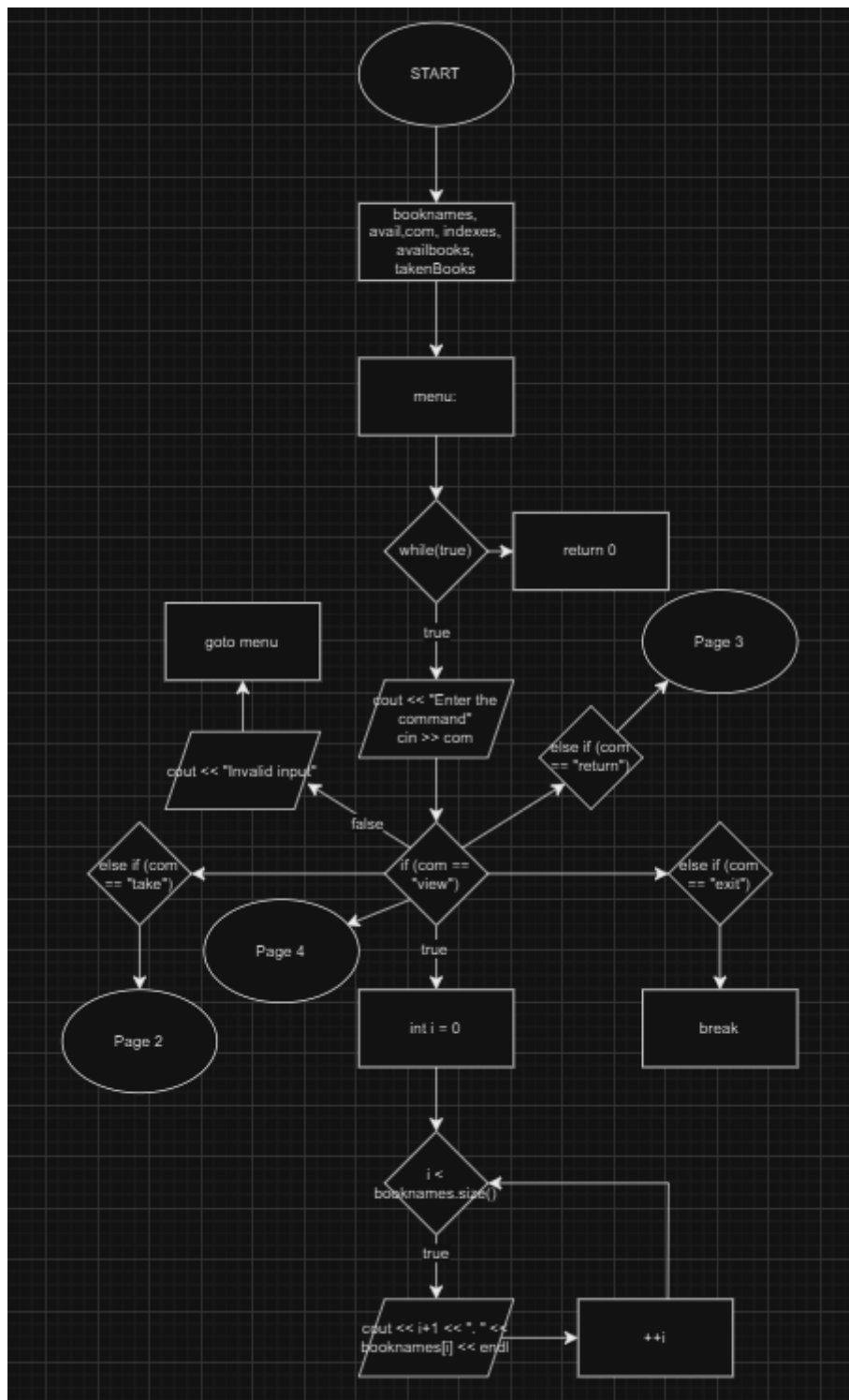
Задача полягала в тому, щоб створити перевантажені функції для знаходження найменшого значення в цілочисельному масиві та найкоротшого слова в рядку.

2) Дизайн та планова оцінка часу виконання завдань:

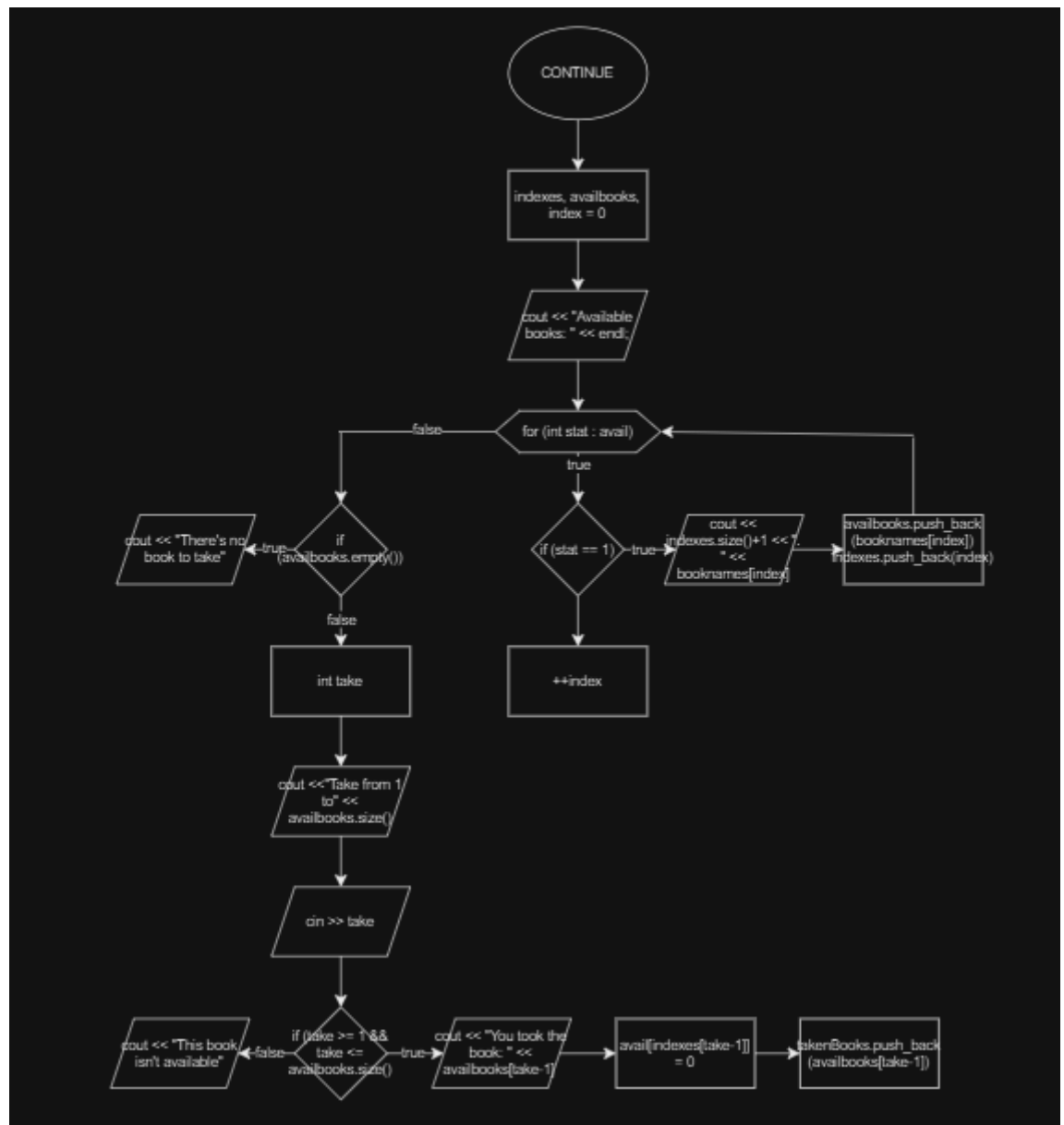
Завдання №1

Спочатку думав витратити на написання програми та блок-схеми максимум 3 години.

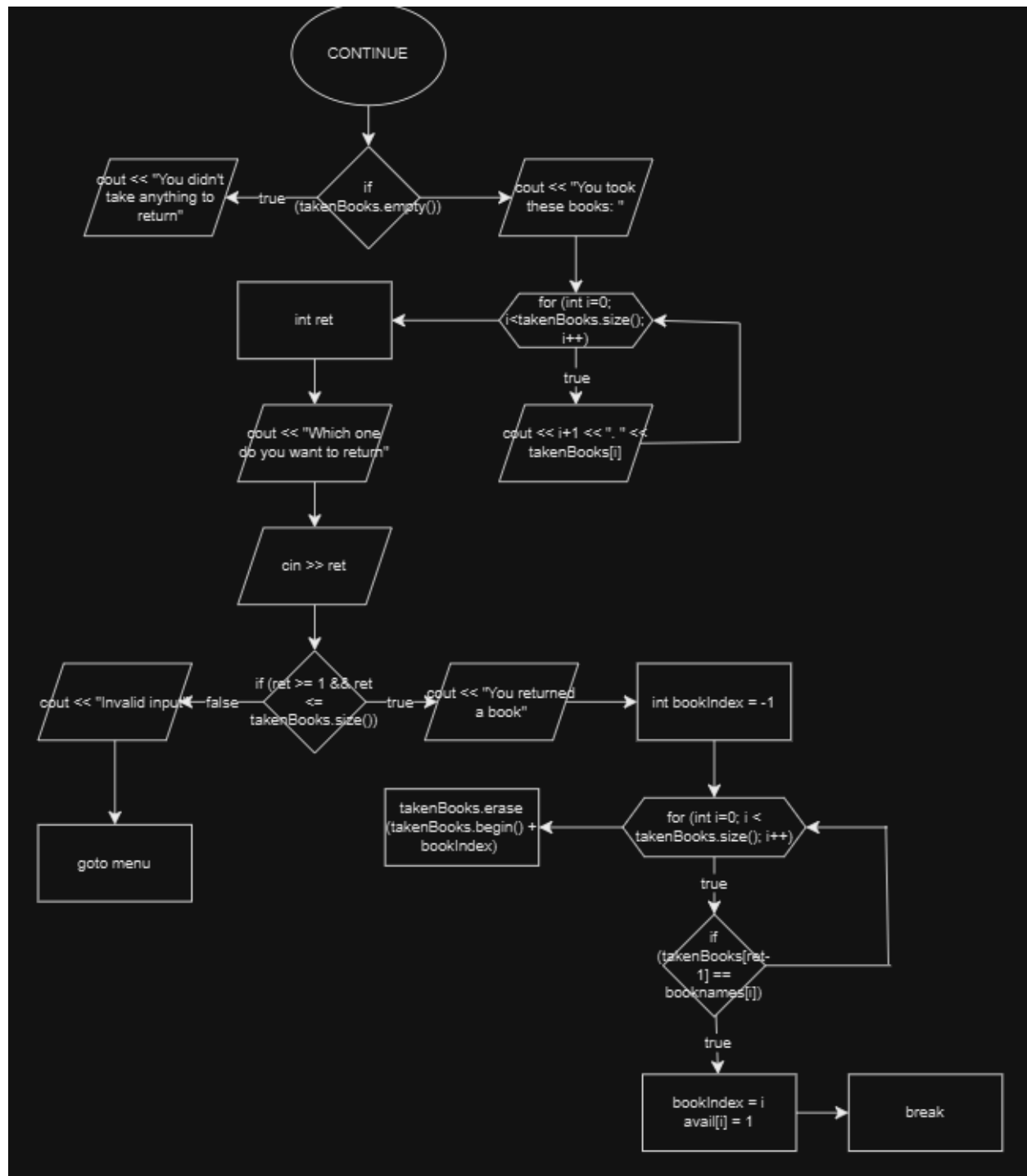
Початок програми:



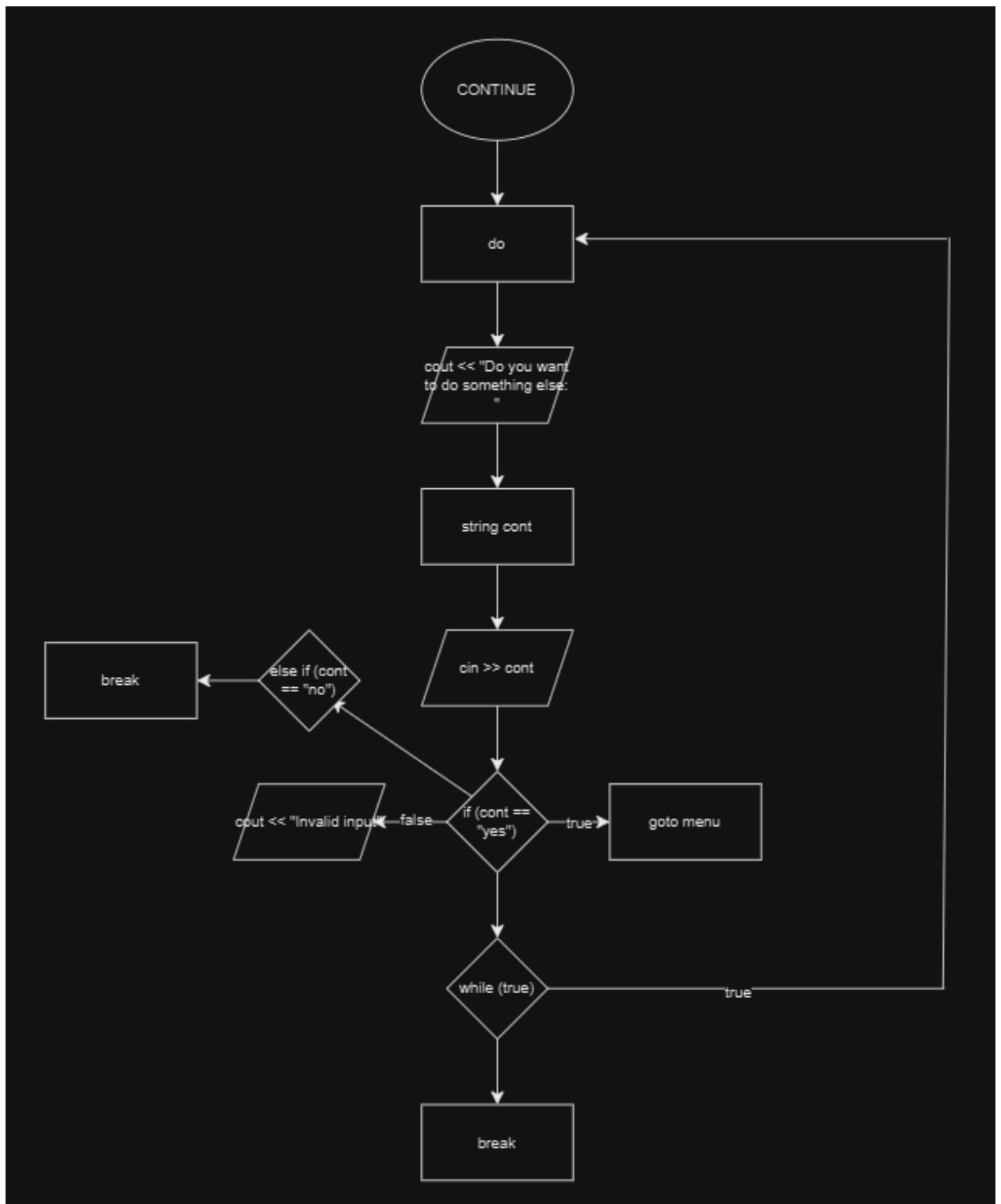
Продовження програми:



Продовження:

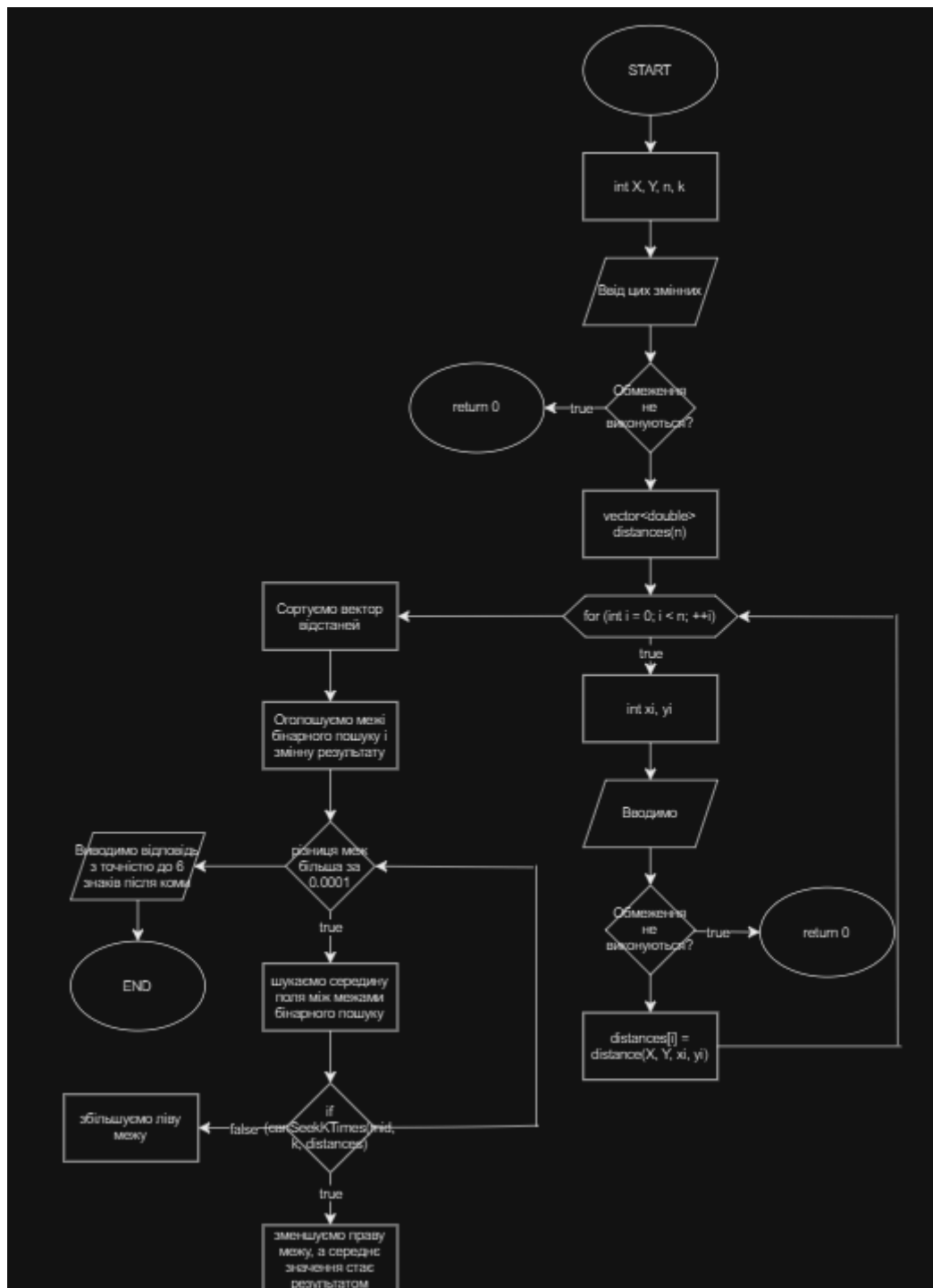


Продовження:



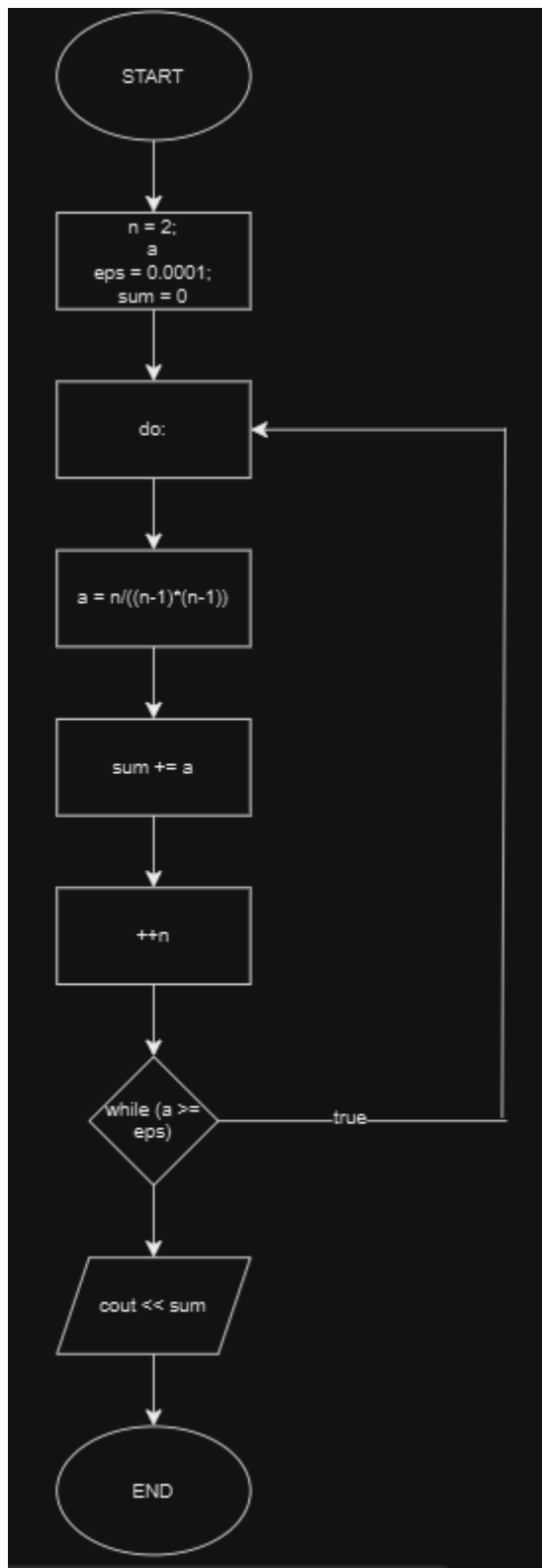
Завдання №2

На це завдання я планував витратити 2 години разим із діаграмою.



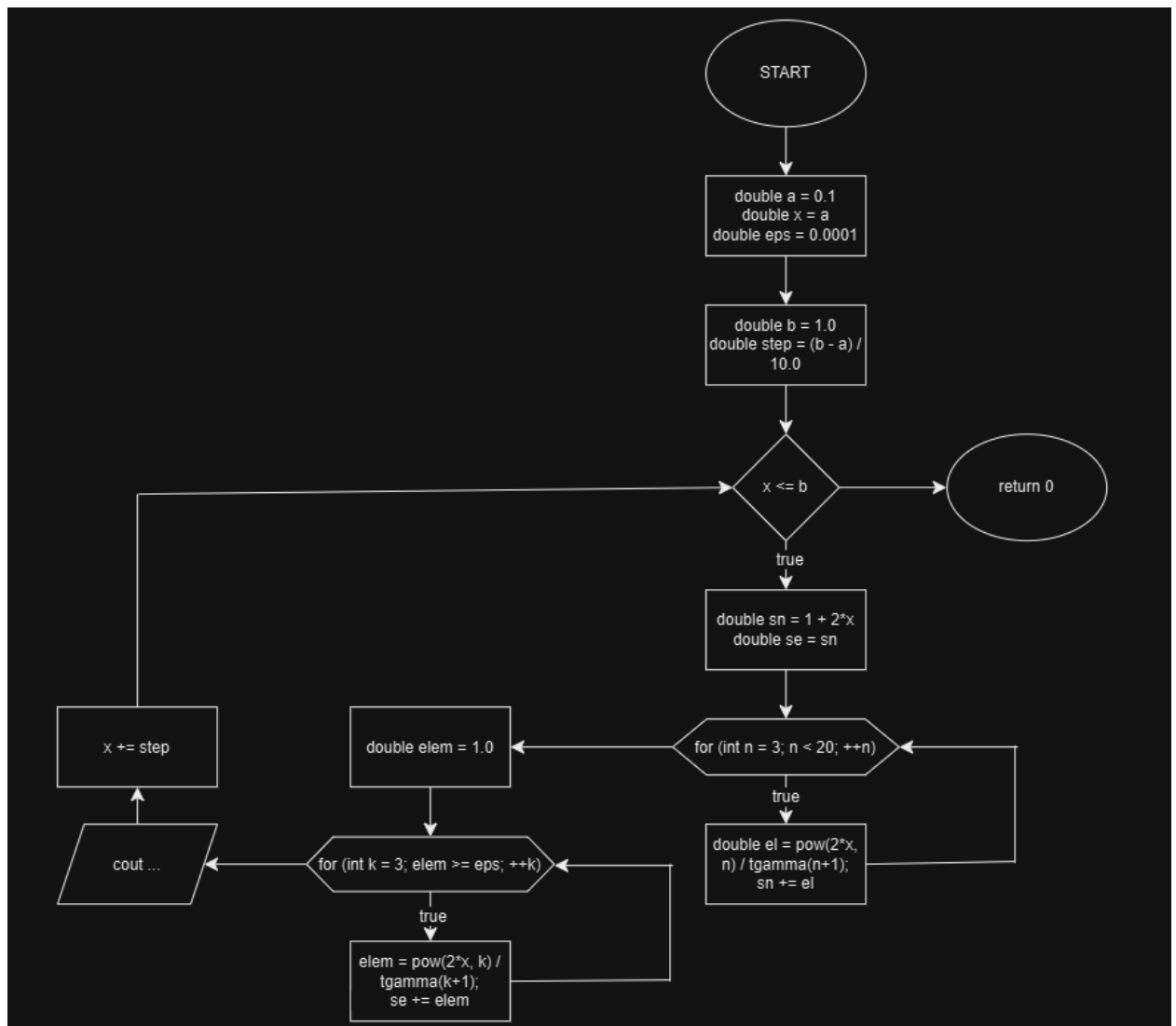
Завдання №3

На це завдання планував витратити не більше 0.5 години.



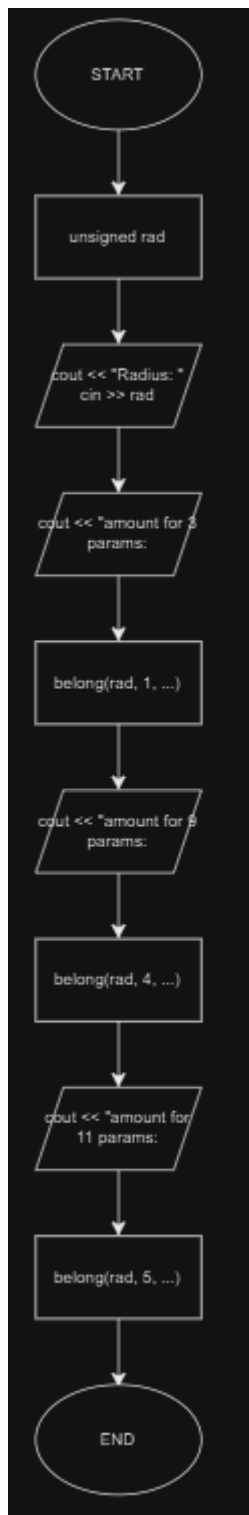
Завдання №4

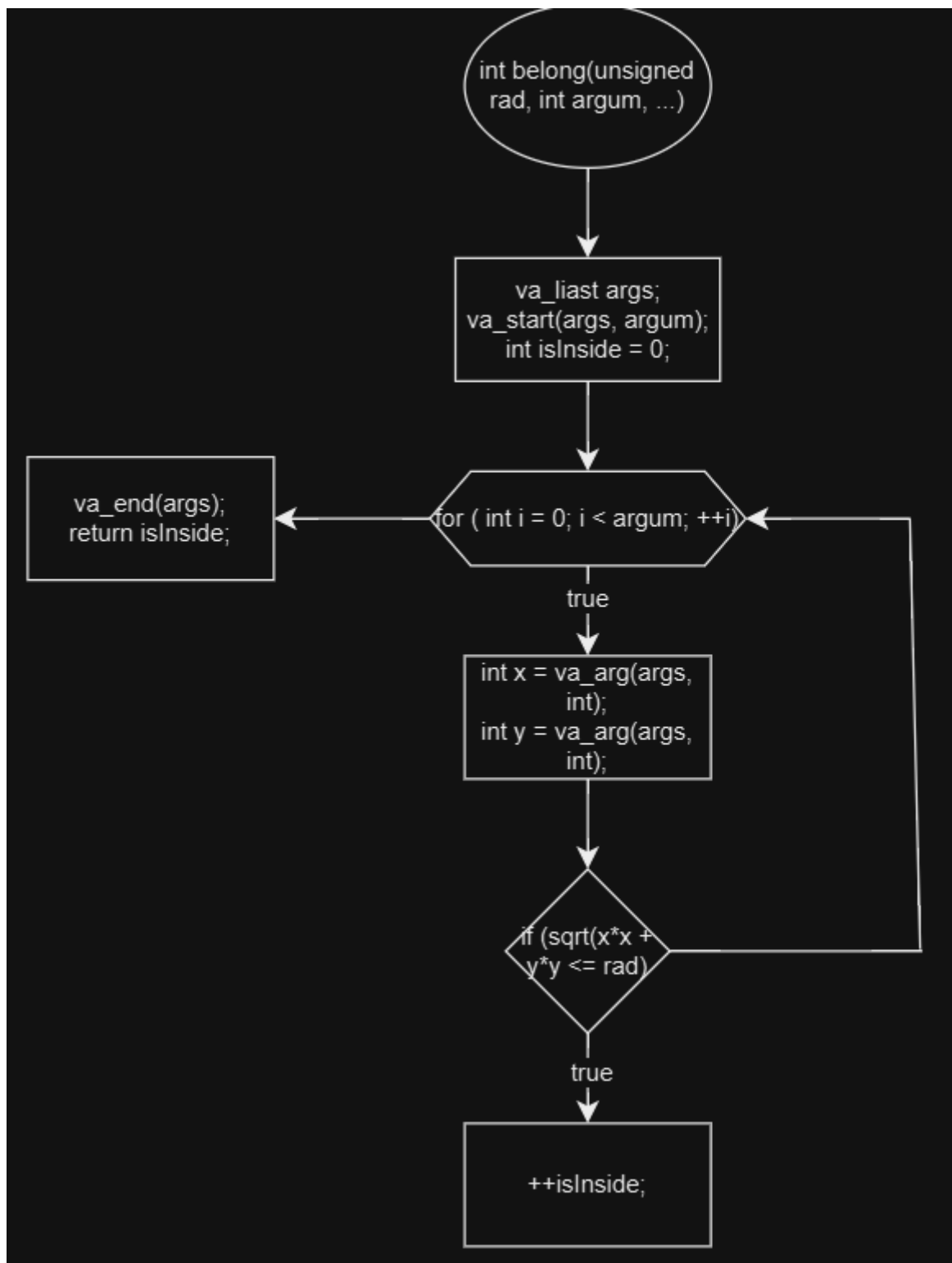
На це завдання я орієнтувався витратити до пів години.



Завдання №5

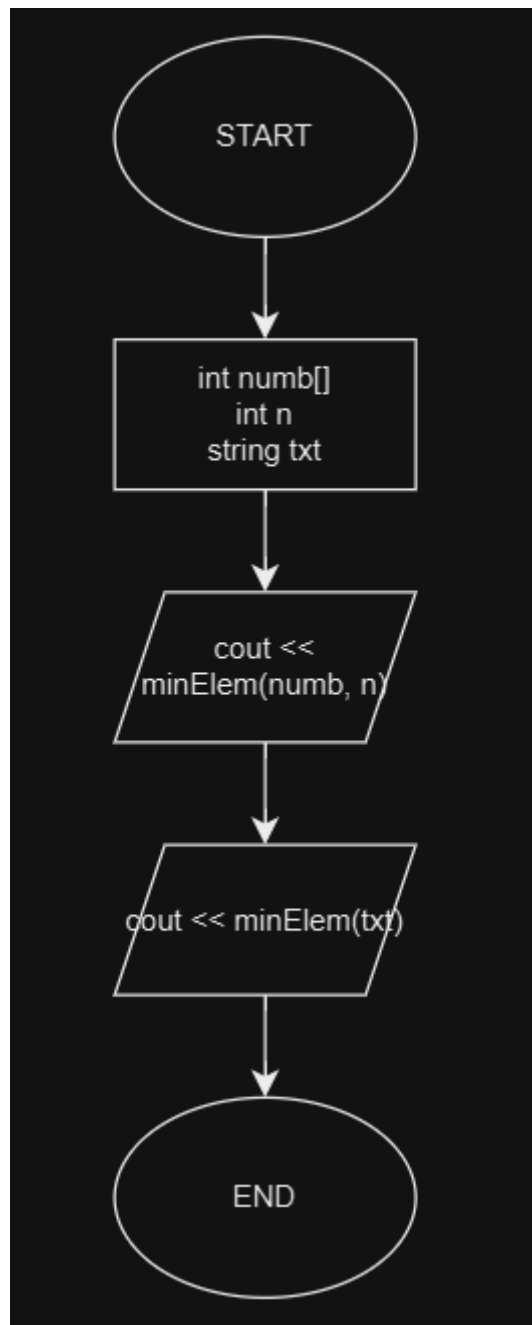
На це завдання думав витратити 45 хвилин.

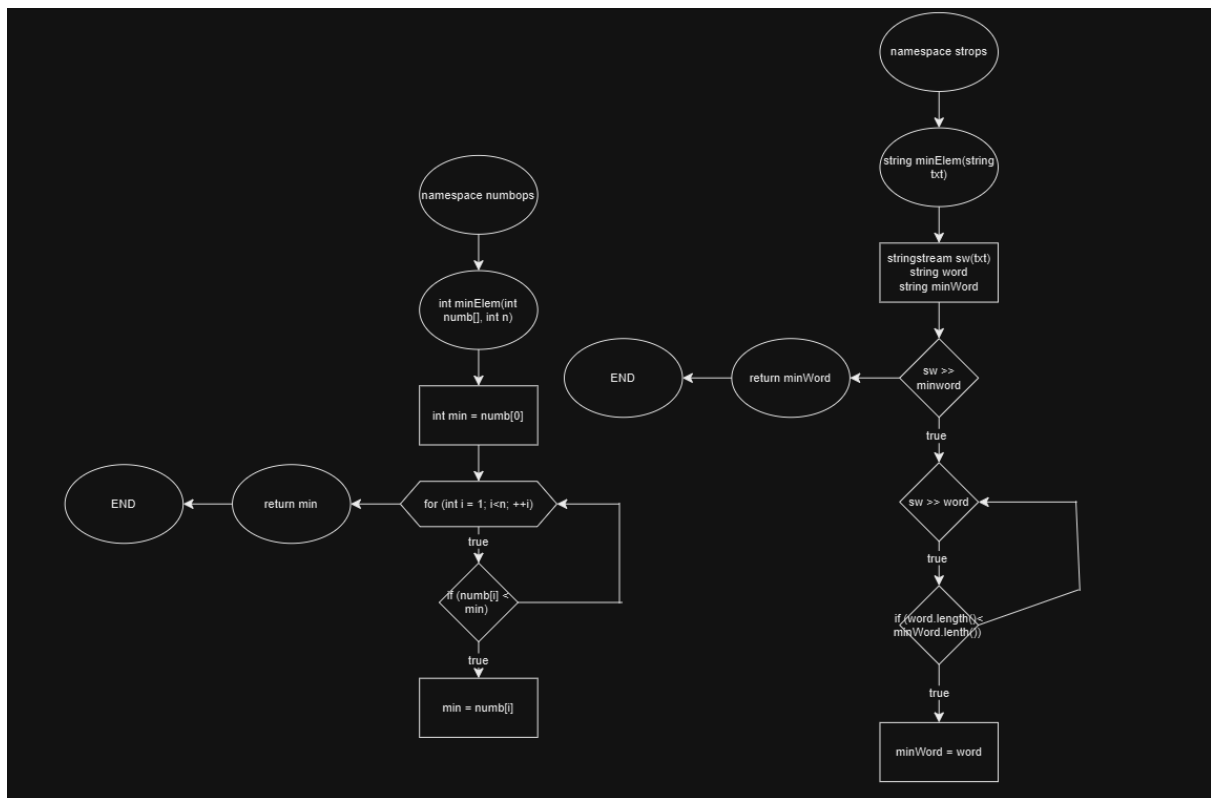




Завдання №6

На це завдання орієнтувався витратити 40 хвилин.





3) Код програм з посиланням на зовнішні ресурси:

Завдання №1


```

#include <iostream>
#include <vector>
#include <string>

using namespace std;

int main() {
    vector<string> booknames = {"1984", "The A.B.C. Murders", "Invisible Monsters", "Six of Crows", "Norwegian Wood"};
    vector<int> avail = {1, 0, 1, 0, 1};
    string com;
    vector<int> indexes;
    vector<string> availbooks;
    vector<string> takenBooks;

    menu:
    while (true) {
        cout << "Hello, what would you like to do? (view/take/return/exit) - ";
        cin >> com;

        if (com == "view") {
            for (int i = 0; i < booknames.size(); ++i) {
                cout << i + 1 << ". " << booknames[i] << endl;
            }
        } else if (com == "take") {
            vector<int> indexes;
            vector<string> availbooks;
            int index = 0;
            cout << "Available books:" << endl;
            for (int stat : avail) {
                if (stat == 1) {
                    cout << indexes.size() + 1 << ". " << booknames[index] << endl;
                    availbooks.push_back(booknames[index]);
                    indexes.push_back(index);
                }
                ++index;
            }

            if (availbooks.empty()) {
                cout << "Sorry, there's no book to take at the point." << endl;
            } else {
                int take;
                cout << "Which one do you want to take (from 1 to " << availbooks.size() << "): ";
                cin >> take;
                if (take >= 1 && take <= availbooks.size()) {
                    cout << "Great, you just took the book: " << availbooks[take-1] << endl;
                    avail[indexes[take-1]] = 0;
                    takenBooks.push_back(availbooks[take-1]);
                } else {
                    cout << "This book is not available." << endl;
                }
            }
        } else if (com == "exit") {
            break;
        }
    }
}

```

```

    } else if (com == "return") {
        if (takenBooks.empty()) {
            cout << "You didn't take anything to return" << endl;
        } else {
            cout << "You took these books: " << endl;
            for (int i = 0; i < takenBooks.size(); i++) {
                cout << i + 1 << ". " << takenBooks[i] << endl;
            }

            int ret;
            cout << "Which one do you want to return? ";
            cin >> ret;

            if (ret >= 1 && ret <= takenBooks.size()) {
                cout << "You just returned a book" << endl;
                int bookIndex = -1;
                for (int i = 0; i < takenBooks.size(); i++) {
                    if (takenBooks[ret-1] == booknames[i]) {
                        bookIndex = i;
                        avail[i] = 1;
                        break;
                    }
                }
                takenBooks.erase(takenBooks.begin() + bookIndex);
            } else {
                cout << "Invalid input" << endl;
                goto menu;
            }
        }
    } else {
        cout << "Invalid input" << endl;
        goto menu;
    }
}

do {
    cout << "Do you want to do something else?(yes/no):" << endl;
    string cont;
    cin >> cont;
    if (cont == "yes" || cont == "y") {
        goto menu;
    } else if (cont == "no" || cont == "n") {
        break;
    } else {
        cout << "Invalid input" << endl;
    }
} while (true);
break;
}

return 0;
}

```

Завдання №2

```

#include <iostream>
#include <vector>
#include <cmath>
#include <algorithm>

using namespace std;

double distance(double x1, double y1, double x2, double y2) {
    return sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
}

bool canSeeKTimes(double r, int k, const vector<double>& distances) {
    int count = 0;
    for (double d : distances) {
        if (d <= r) {
            count++;
        }
        if (count >= k) {
            return true;
        }
    }
    return false;
}

int main() {
    int X, Y;
    int n, k;
    cin >> X >> Y;
    cin >> n >> k;

    if (X < 0 || X > 1e9 || Y < 0 || Y > 1e9) {
        return 0;
    }

    if (n < 1 || n > 100 || k < 1 || k > n) {
        return 0;
    }

    vector<double> distances(n);
    for (int i = 0; i < n; i++) {
        int xi, yi;
        cin >> xi >> yi;
        if (xi < 0 || xi > 1e9 || yi < 0 || yi > 1e9) {
            return 0;
        }
        distances[i] = distance(X, Y, xi, yi);
    }

    sort(distances.begin(), distances.end());
}

```

```

double left = 0.0, right = distances.back();
double answer = right;

while (right - left > 1e-4) {
    double mid = (left + right) / 2.0;
    if (canSeeKTimes(mid, k, distances)) {
        answer = mid;
        right = mid;
    } else {
        left = mid;
    }
}

cout.precision(6);
cout << fixed << answer << endl;

return 0;

```

Завдання №3

```

#include <iostream>
using namespace std;

int main() {
    double n = 2.0;
    double a;
    double eps = 0.0001;
    double sum = 0.0;
    do {
        a = n / ((n-1.0)*(n-1.0));
        sum += a;
        ++n;
    } while (a >= eps);
    cout << sum;
    return 0;
}

```

Завдання №4

```

#include <iostream>
#include <cmath>
using namespace std;

int main() {
    double a = 0.1;
    double x = a;
    double eps = 0.0001;
    double b = 1.0;
    double step = (b - a) / 10.0;

    for (; x <= b; x += step) {
        double sn = 1.0 + 2.0 * x;
        double se = sn;

        for (int n = 3; n < 20; ++n) {
            double el = pow(2 * x, n) / tgamma(n + 1.0);
            sn += el;
        }

        double elem = 1.0;
        for (int k = 3; elem >= eps; ++k) {
            elem = pow(2.0 * x, k) / tgamma(k + 1);
            se += elem;
        }

        cout << "X = " << x << ": SN = " << sn << ", SE = " << se << ", Y = e ^ (2x)" << endl;
    }

    return 0;
}

```

Завдання №5

```

#include <iostream>
#include <cstdlib>
#include <cmath>
using namespace std;

int belong(unsigned rad, int argum, ...) {
    va_list args;
    va_start(args, argum);
    int isInside = 0;
    for (int i = 0; i < argum; i++) {
        int x = va_arg(args, int);
        int y = va_arg(args, int);
        if (sqrt(x*x + y*y) <= rad) {
            ++isInside;
        }
    }
    va_end(args);
    return isInside;
}

int main() {
    unsigned rad;
    cout << "Radius: ";
    cin >> rad;
    cout << "amount of dots in circle(3 params): " << belong(rad, 1, 3, 5)
    << endl;
    cout << "amount of dots in circle(9 params): " << belong(rad, 4, 3, -6, 7, -2, 1, 3, 9, -2)
    << endl;
    cout << "amount of dots in circle(11 params): " << belong(rad, 5, 9, -8, 7, 1, -3, -4, 4, 6, 2, 10)
    << endl;
    return 0;
}

```

Завдання №6

```
#include <iostream>
#include <string>
#include <sstream>
using namespace std;

namespace numbops {
    int minElem(int numb[], int n) {
        int min = numb[0];

        for (int i = 1; i < n; ++i) {
            if (numb[i] < min) {
                min = numb[i];
            }
        }
        return min;
    }
}

namespace strops {
    string minElem(string txt) {
        stringstream sw(txt);
        string word;
        string minWord;

        if (sw >> minWord) {
            while (sw >> word) {
                if (word.length() < minWord.length()) {
                    minWord = word;
                }
            }
        }
        return minWord;
    }
}

int main() {
    int numb[] = {-23, 34, 3, 0, 145, -56};
    int n = 6;
    string txt = "Can you pass me salt shaker, please";

    cout << "Мінімальне число: " << numbops::minElem(numb, n) << endl;
    cout << "Мінімальне слово: " << strops::minElem(txt) << endl;

    return 0;
}
```

```

#include <iostream>
#include <string>
#include <sstream>
using namespace std;

int minElemNumbers(int numb[], int n) {
    int min = numb[0];

    for (int i = 1; i < n; ++i) {
        if (numb[i] < min) {
            min = numb[i];
        }
    }
    return min;
}

string minElemString(string txt) {
    stringstream sw(txt);
    string word;
    string minWord;

    if (sw >> minWord) {
        while (sw >> word) {
            if (word.length() < minWord.length()) {
                minWord = word;
            }
        }
    }
    return minWord;
}

int main() {
    int numb[] = {-23, 34, 3, 0, 145, -56};
    int n = 6;
    string txt = "Can you pass me salt shaker, please";

    cout << "Мінімальне число: " << minElemNumbers(numb, n) << endl;
    cout << "Мінімальне слово: " << minElemString(txt) << endl;

    return 0;
}

```

4) Результати виконання завдань, тестування та фактично
затрачений час

Завдання №1


```
Hello, what would you like to do? (view/take/return/exit) - take
Available books:
1. 1984
2. Invisible Monsters
3. Norwegian Wood
Which one do you want to take (from 1 to 3): 1
Great, you just took the book: 1984
Do you want to do something else?(yes/no):
y
Hello, what would you like to do? (view/take/return/exit) - return
You took these books:
1. 1984
Which one do you want to return? 1
You just returned a book
Do you want to do something else?(yes/no):
no
PS D:\> █
```

Тут я взяв і потім повернув книжку.

Витратив 1 день.

Завдання №2

```
5 6
4 2
3 4
8 11
6 5
17 25
2.828432
1.5
```

Витратив на завдання близько 1 години.

Завдання №3

```
11.4325
PS D:\> █
```

На це завдання пішло 25 хвилин.

Завдання №4

```
X = 0.1: SN = 1.2014, SE = 1.2014, Y = e ^ (2x)
X = 0.19: SN = 1.39008, SE = 1.39008, Y = e ^ (2x)
X = 0.28: SN = 1.59387, SE = 1.59387, Y = e ^ (2x)
X = 0.37: SN = 1.82214, SE = 1.82213, Y = e ^ (2x)
X = 0.46: SN = 2.08609, SE = 2.08609, Y = e ^ (2x)
X = 0.55: SN = 2.39917, SE = 2.39916, Y = e ^ (2x)
X = 0.64: SN = 2.77744, SE = 2.77744, Y = e ^ (2x)
X = 0.73: SN = 3.24016, SE = 3.24015, Y = e ^ (2x)
X = 0.82: SN = 3.81037, SE = 3.81036, Y = e ^ (2x)
X = 0.91: SN = 4.51566, SE = 4.51566, Y = e ^ (2x)
X = 1: SN = 5.38906, SE = 5.38905, Y = e ^ (2x)
PS D:\> █
```

Витратив на завдання приблизно 40 хвилин.

Завдання №5

```
Radius: 8
amount of dots in circle(3 params): 1
amount of dots in circle(9 params): 3
amount of dots in circle(11 params): 3
PS D:\> █
```

Результат для радіусу 8.

Завдання №6

```
Мінімальне число: -56
Мінімальне слово: me
PS D:\> █
```

Командна робота



Висновки:

У ході лабораторної роботи я ознайомився з основними принципами роботи циклів, включаючи вкладені цикли та способи завершення їх виконання, у мові програмування C++.

Я навчився створювати та використовувати функції, розуміти простір імен, реалізовувати перевантаження функцій і функції з змінною кількістю параметрів (еліпсис). Також я дослідив застосування рекурсії для розв'язання завдань, що потребують багаторазового повторення операцій.

Крім того, я вивчив можливості та застосування вбудованих функцій для спрощення програмного коду. Лабораторна робота дозволила мені закріпити знання про основи структурного програмування, поглибити розуміння роботи функцій і циклів у C++ та отримати практичний досвід у створенні ефективних програм.

[Pull request](#)

