

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:

Студент групи ІІІ-11
Фарина Арсеній Петрович

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:

- Тема №1: Введення в Цикли та їх Види в C++.
- Тема №2: Управління Виконанням Циклів.
- Тема №3: Вкладені Цикли.
- Тема №4: Основи Функцій у C++
- Тема №5: Перевантаження Функцій та Простір Імен.
- Тема №6: Розширені Можливості Функцій.
- Тема №7: Вбудовані Функції в C++

1) Індивідуальний план опрацювання теорії:

- Тема №1: Введення в цикли та їх види в C++:
 - o Джерела інформації:
 - Статті.
<https://www.youtube.com/watch?v=zBtcqNdiRf4&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=31>
<https://www.youtube.com/watch?v=ckJtOMcIxyU&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=32>
 - Що опрацьовано:
 - o Значення та роль циклів у програмуванні.
 - o Огляд видів циклів: for, while, do-while.
 - o Синтаксис та основи використання кожного типу циклу.
 - o Приклади базових циклів для різних задач.
- Запланований час на вивчення 1 година.
Витрачений час 1 година.
- Тема №2: Управління виконанням циклів:
 - o Джерела інформації:
 - Статті.
<https://www.youtube.com/watch?v=rj1OLsBKazA&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=36>
<https://www.youtube.com/watch?v=UY295pIdeoQ&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=37>
 - Що опрацьовано:
 - o Застосування операторів break та continue.
 - o Умови завершення циклів.
 - o Передчасне завершення виконання циклу.
 - o Приклади та вправи з управлінням циклами.
- Запланований час на вивчення 30 хвилин.
Витрачений час 30 хвилин.
- Тема №3: Вкладені цикли:
 - o Джерела інформації:
 - Статті.
<https://www.youtube.com/watch?v=mBPHKQx21eE>

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

Запланований час на вивчення 1 година.
Витрачений час 1 година.

- Тема №4 Основи функцій у C++:
 - Джерела інформації:
 - Статті.

<https://www.youtube.com/watch?v=G8P6SvdqU9s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=43>
 - Що опрацьовано
 - Визначення та оголошення функцій.
 - Параметри функцій: передача за значенням і за посиланням.
 - Параметри за замовчуванням.
 - Повернення значень з функцій.
 - Приклади створення та використання функцій.

Запланований час на вивчення 2 години.
Витрачений час 2 години.

- Тема № 5 Перевантаження функцій та простір імен:
 - Джерела інформації:
 - Статті.

<https://www.youtube.com/watch?v=hcYgFCgeZzQ>
 - Що опрацьовано
 - Концепція перевантаження функцій.
 - Правила та приклади перевантаження функцій.
 - Поняття та використання просторів імен.
 - Вкладені простори імен (C++ 17)
 - Роль просторів імен у організації коду.

Запланований час на вивчення 1 година.
Витрачений час 1 година..

- Тема №6: Розширені можливості функцій:
 - Джерела інформації:
 - Статті.

<https://acode.com.ua/urok-15-funktsiyi-i-operator-return/>
https://www.youtube.com/watch?v=_N3zkbnCTw0
 - Що опрацьовано
 - 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.

Запланований час на вивчення 1 година.
Витрачений час 1 година.

- Тема №7: Вбудовані функції в C++:
 - Джерела інформації:
 - Статті.

https://www.youtube.com/watch?v=V_8XRRlus7Y&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=49
 - Що опрацьовано

- Огляд вбудованих функцій у C++.
 - Приклади використання стандартних функцій у програмуванні.
 - Роль вбудованих функцій у спрощенні коду.
 - Практичні завдання для розуміння вбудованих функцій.
- Запланований час на вивчення 1 година.
Витрачений час 1 година.

Виконання роботи:

1. Опрацювання завдання та вимог до програм.

Завдання №1

VNS LAB 2 – TASK 1

Знайти суму цілих додатніх парних чисел, менших 100.

Завдання №2

VNS LAB 3 – TASK 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 ϵ ($\epsilon=0.0001$).

Для порівняння знайти точне значення функції.

2	$y = -\ln \left 2 \sin \frac{x}{2} \right $	$\frac{\pi}{5} \leq x \leq \frac{9\pi}{5}$	40	$S = \cos x + \frac{\cos 2x}{2} + \dots + \frac{\cos nx}{n}$
---	--	--	----	--

Завдання №3

VNS LAB 7 – TASK 1

Написати функцію `mult` зі змінною кількістю параметрів, що знаходить добуток чисел типу `float`. Написати викликаючу функцію `main`, що звертається до функції `mult` не менше трьох разів з кількістю параметрів 3, 7, 11.

Завдання №4

VNS LAB 7 – TASK 2

Написати перевантажені функції й основну програму, що їх викликає.

2.

а) для додавання дійсних чисел;

б) для додавання комплексних чисел.

Завдання №5

Class Practice work

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

Деталі

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) запитуєте 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 algotester

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

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

і-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_1 , другий почнеться в l_1 і закінчиться в $l_1 + l_2$ 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.

Вхідні дані

У першому рядку задане ціле число n — кількість компаній.

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

Вихідні дані

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

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

2. Дизайн та планувальна оцінка часу виконання завдань:

Програма №1

- Блок-схема
- Важливі деталі для реалізації програми
Використати цикл for для того щоб пройтися по числах від 0 до 100 і в середині циклу прописати if, щоб додавали тільки ті числа, які при діленні на два не дають остачі, тобто парні числа.
- Плановий час на реалізацію 30 хвилин.

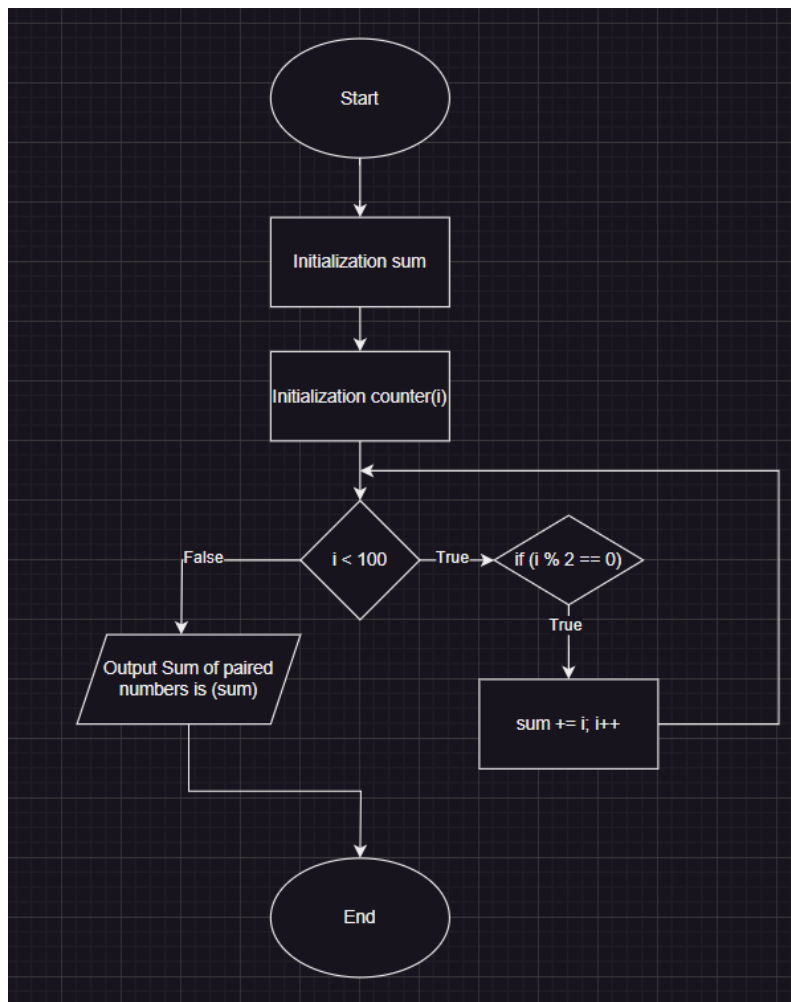


Рисунок 2.1. Блок-схема до програми 1

Програма №2

- Блок-схема
 - Важливі деталі для реалізації програми
- Написати функції для обрахунку значення з точністю ϵ ($\epsilon=0.0001$), функцію для значення суми SN, і функцію SE - для заданої точності, а також виводити на екран X- значення параметра; SN- значення суми для заданого n; SE- значення суми для заданої точності; Y-точне значення функції.
- Плановий час на реалізацію 2.5 години.

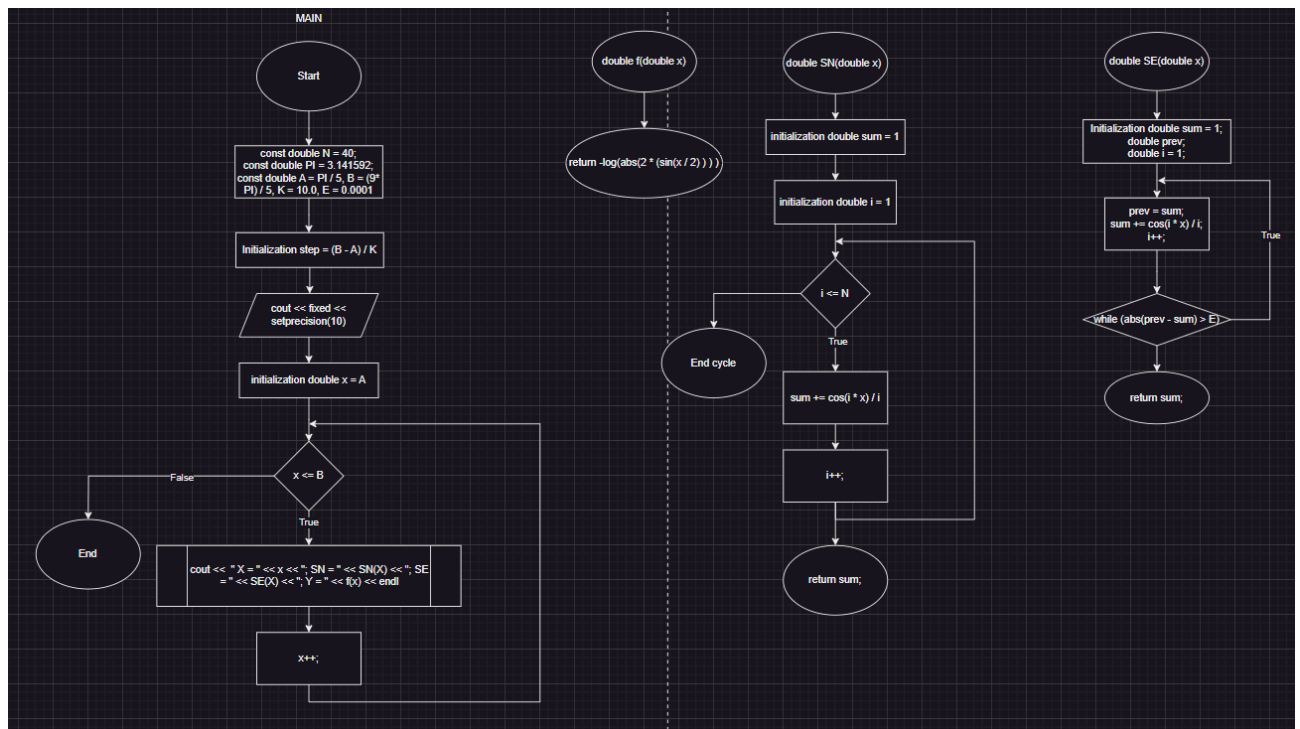


Рисунок 2.2. Блок-схема до програми 2

Програма №3

- Блок-схема
- Важливі деталі для реалізації програми
- Використати перевантажену функцію mult, і навчитися працювати з бібліотекою cstdarg.
- Плановий час на реалізацію 1.5 години.

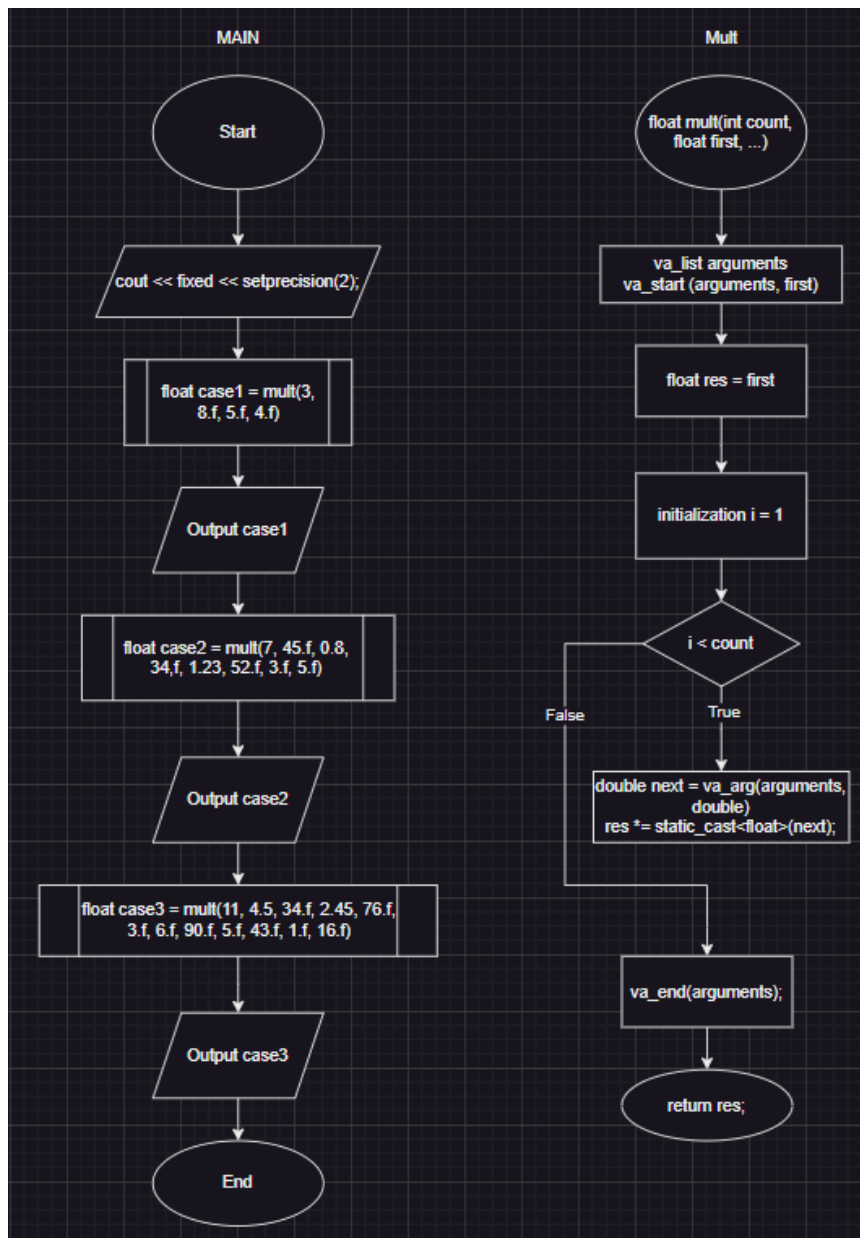


Рисунок 2.3. Блок-схема до програми 3

Програма №4

- Блок-схема
- Важливі деталі для реалізації програми
- Написати перевантажені функції та викликати їх у головній функції
- Плановий час на реалізацію 2 години.

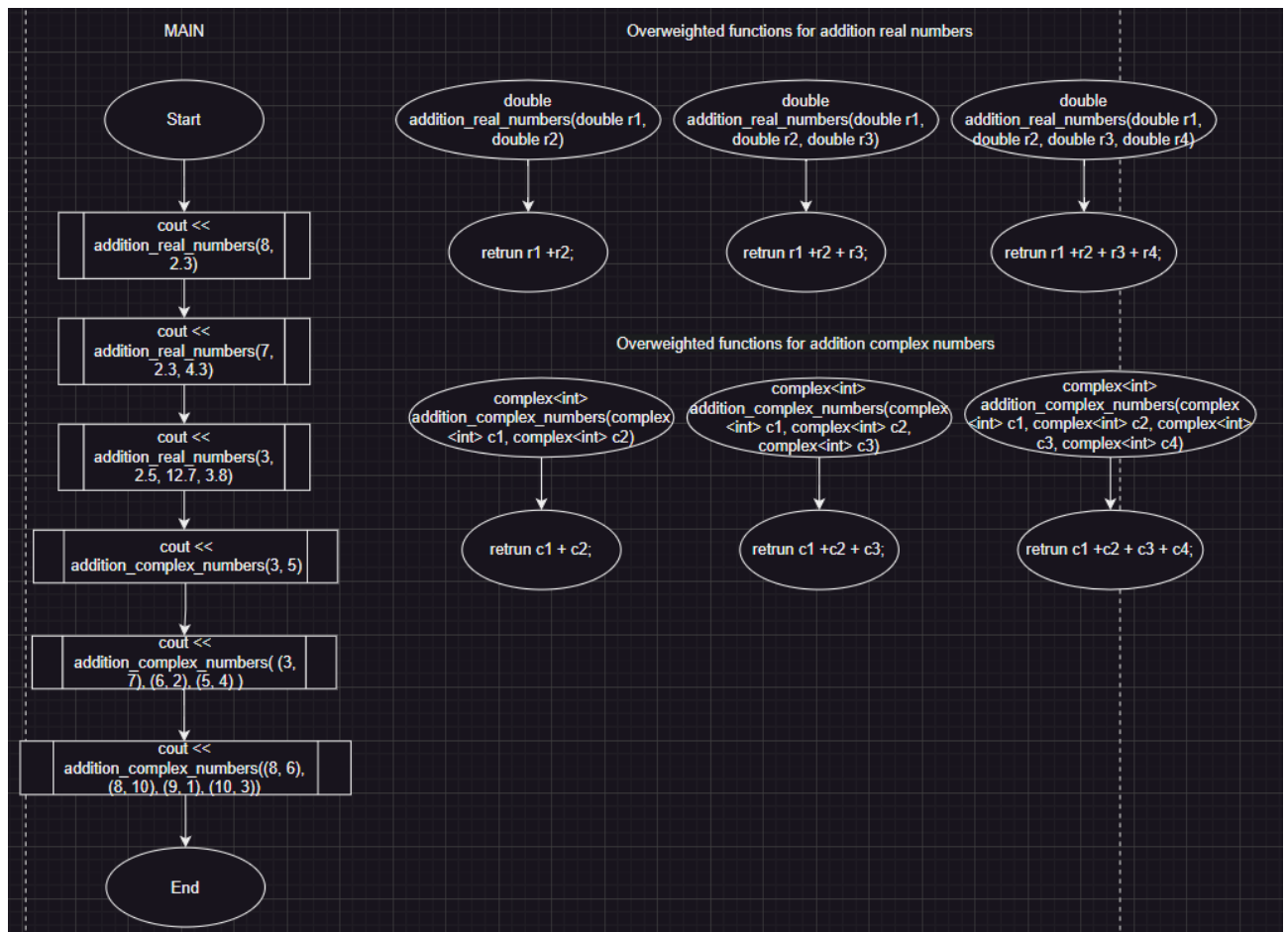
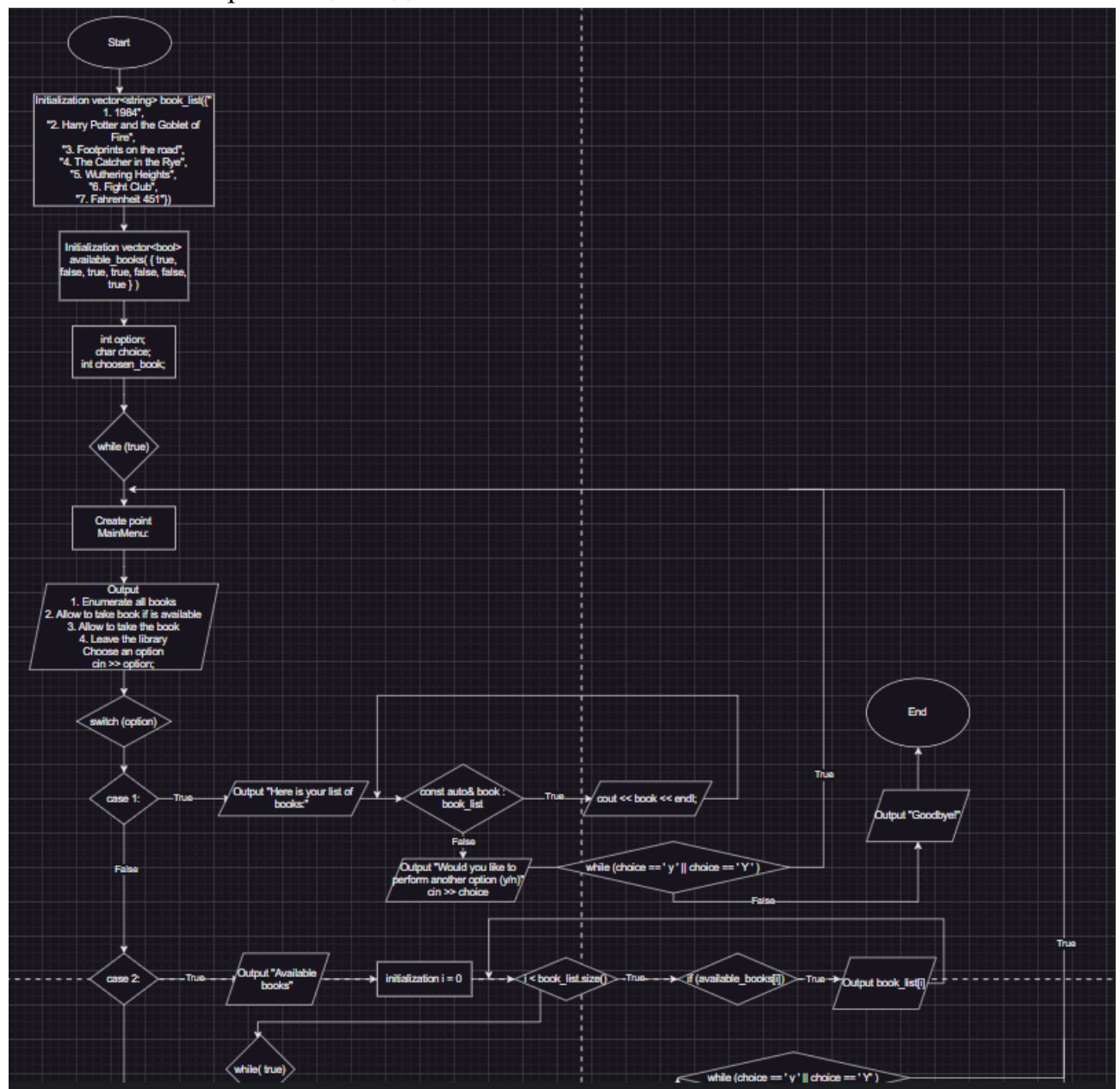


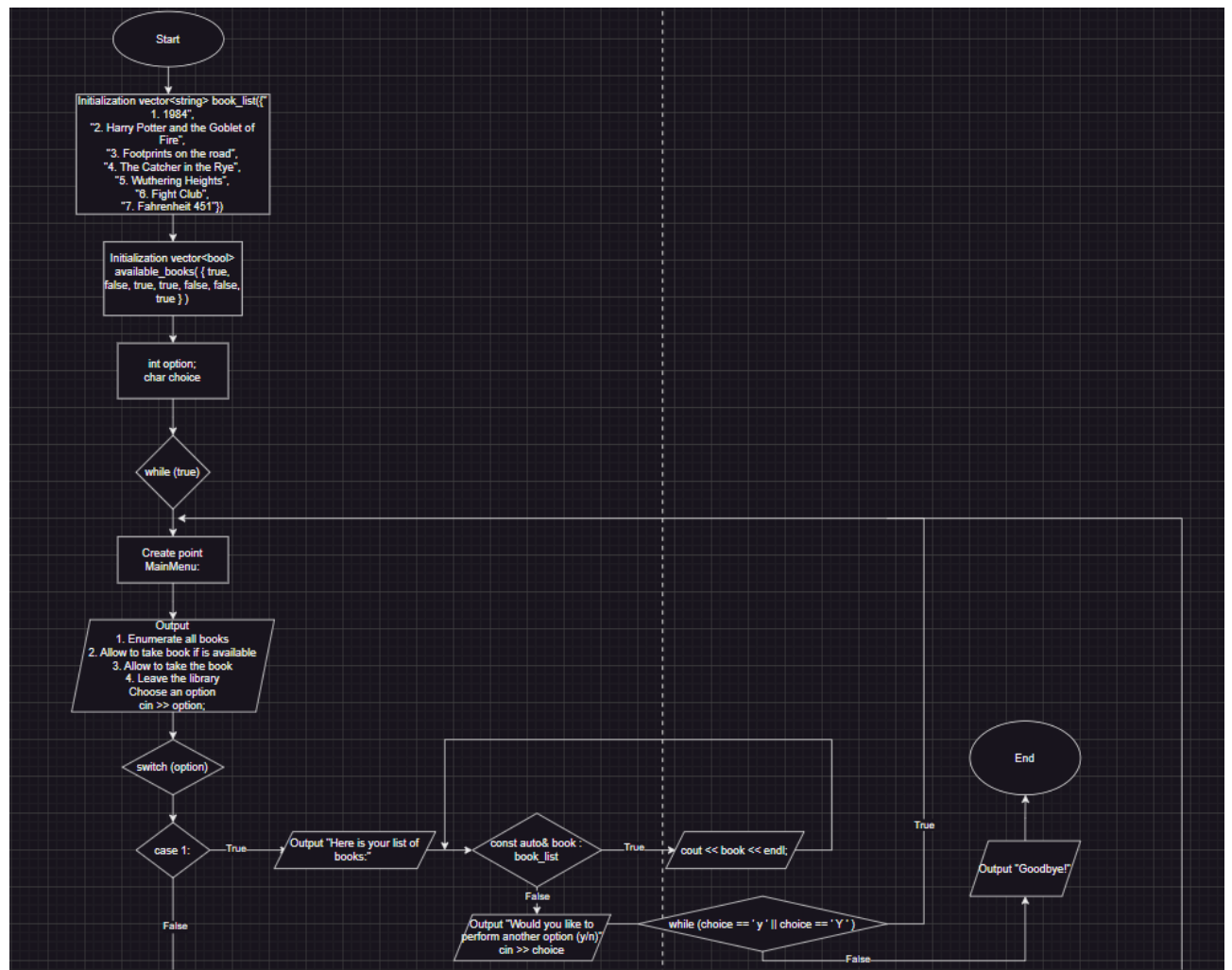
Рисунок 2.4. Блок-схема до програми 4

Програма №5

- Блок-схема
- Важливі деталі для реалізації програми
-

- Плановий час на реалізацію 1 день





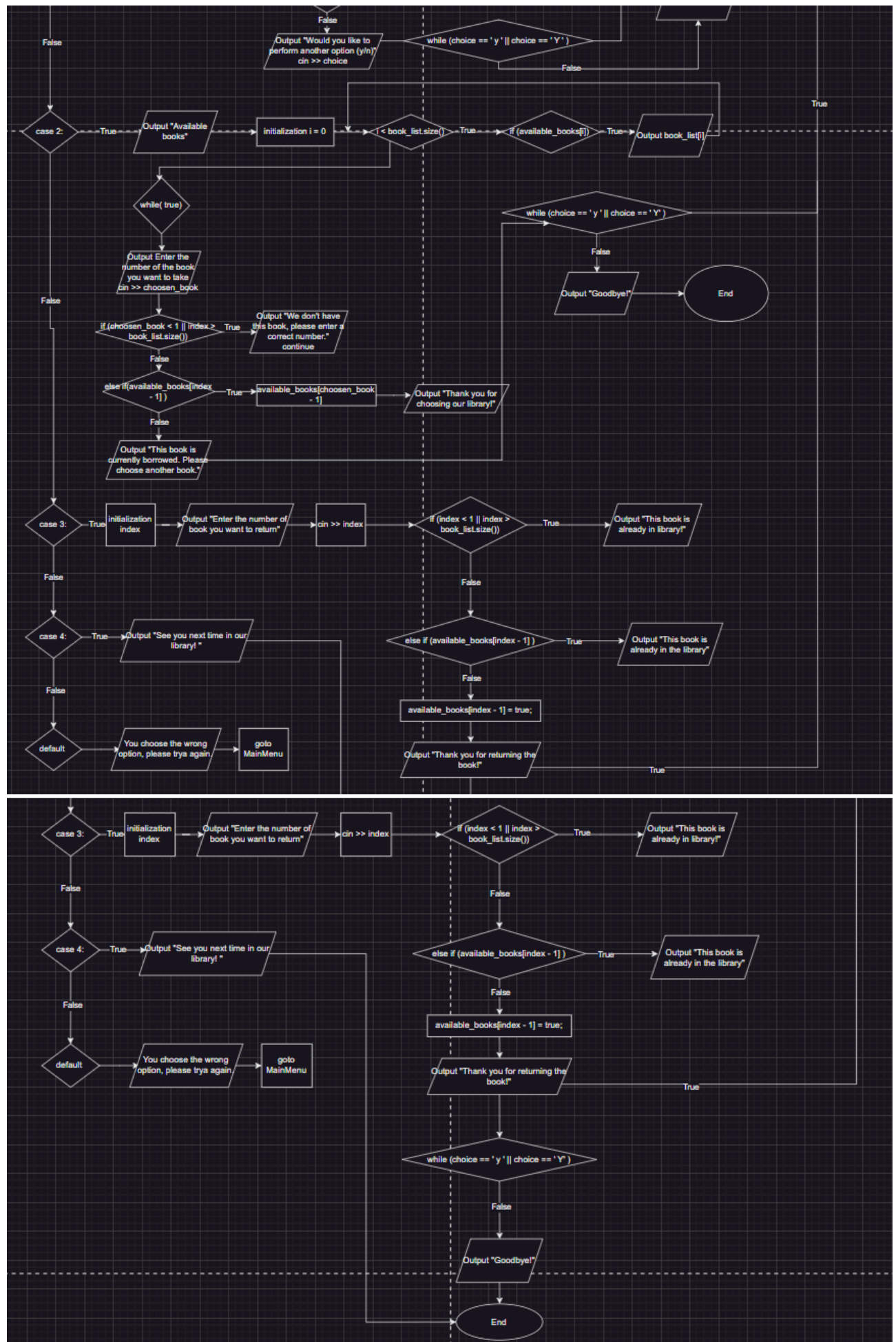


Рисунок 2.5. Блок-схема до програми №5

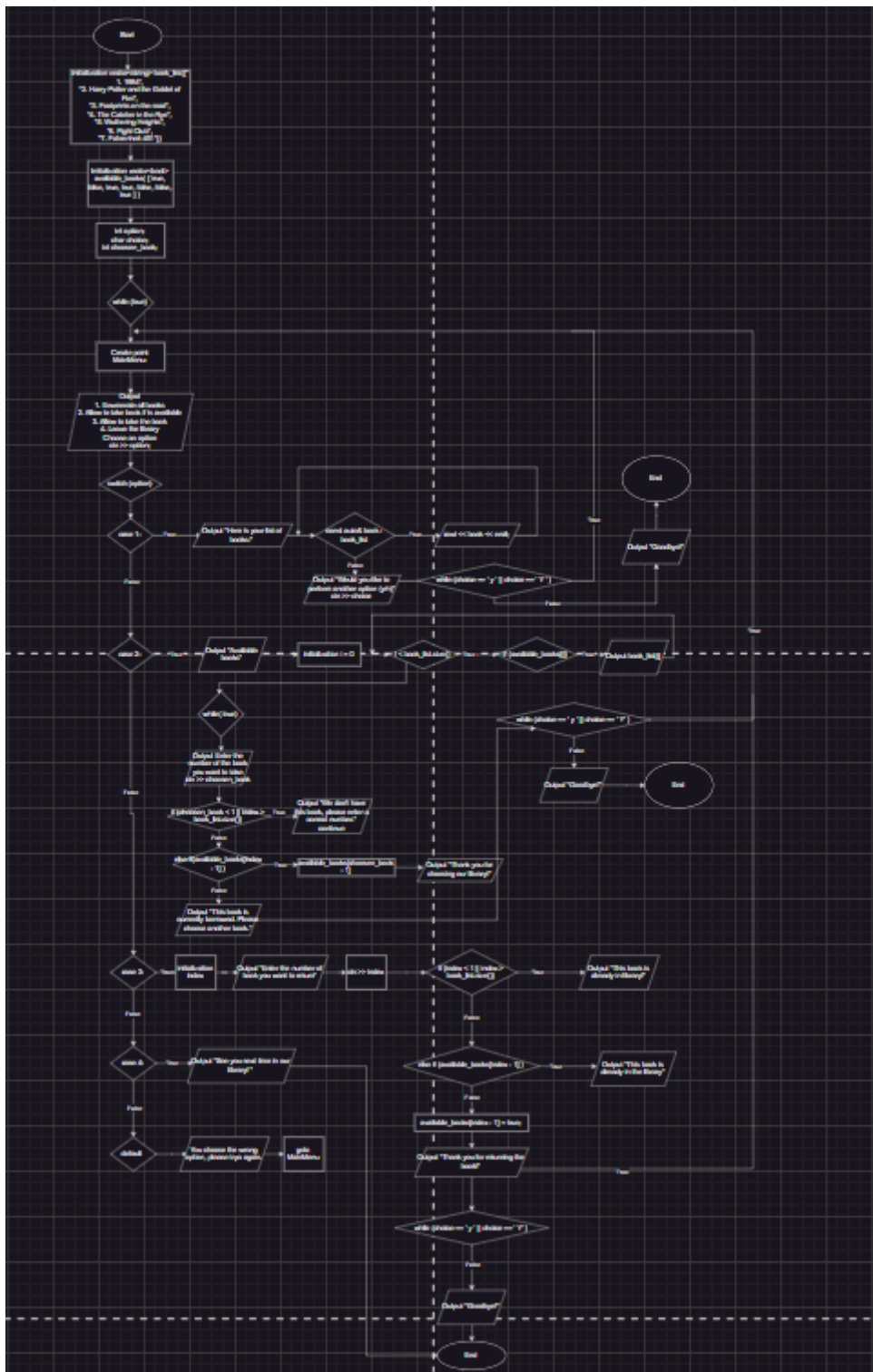


Рисунок 2.6. Загальна блок-схема до програми №5

Програма №6

- Блок-схема
- Важливі деталі для реалізації програми
- Використати цикл для перевірки елементів у векторі.
- Плановий час на реалізацію 1 день

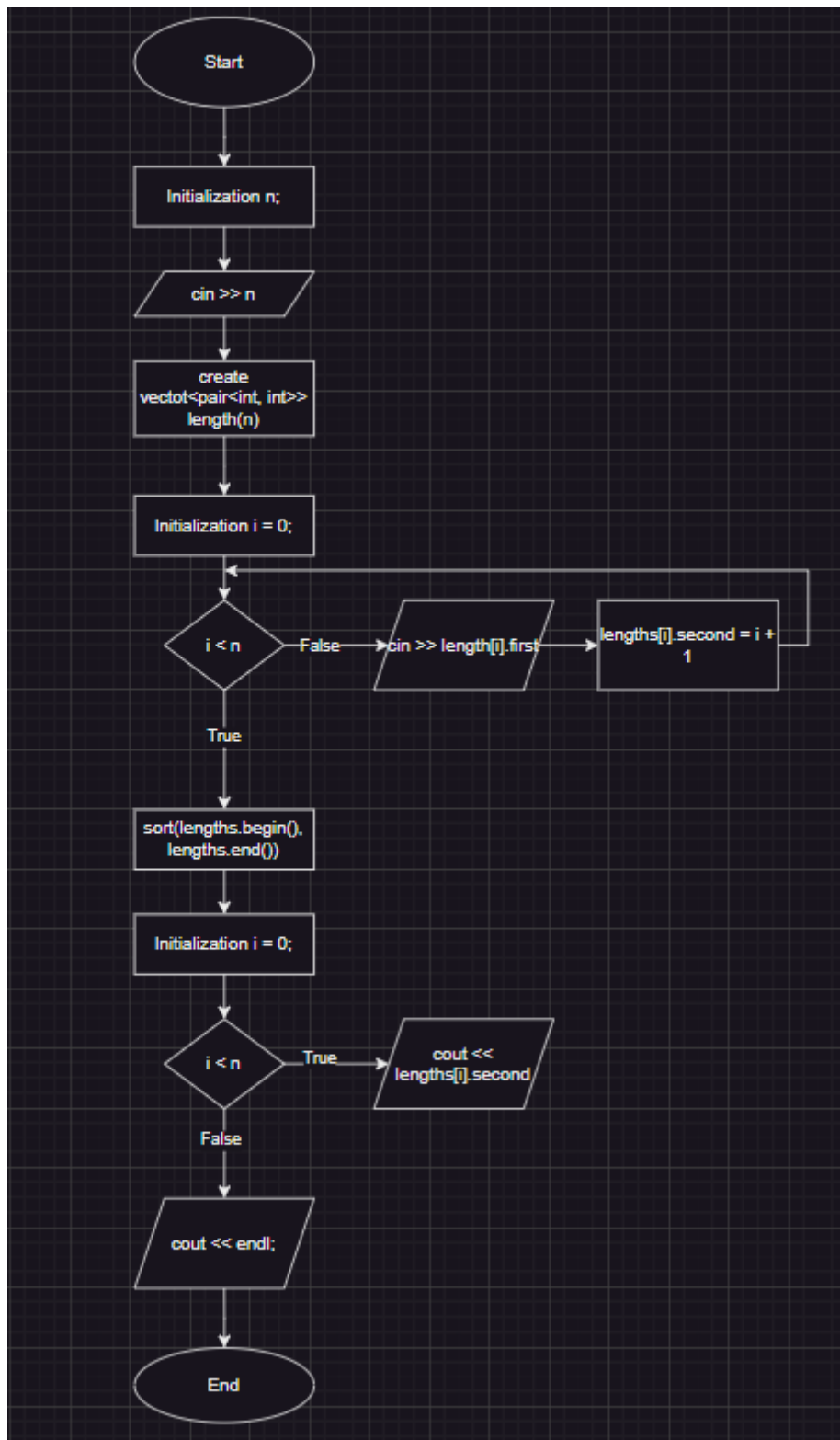



Рисунок 2.7. Блок-схема до програми №6

3. Код програм з посиланням на зовнішні ресурси та фактично затрачений час:

Завдання №1



```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      int sum;
8      for (int i = 0; i < 100; i++){
9          if (i % 2 == 0){
10             sum += i;
11         }
12     }
13
14     cout << "Sum of paired number is: " << sum << endl;
15
16     return 0;
17 }
```

Рисунок 3.1. Код до програми №1

```
Sum of paired number is: 2450
PS D:\ai_programming_playground_2024\ai_11\faryna_arsenii\epic_3>
```

Рисунок 3.2. Приклад виконання програми №1

На початку ми ініціалізуємо змінну `sum` в яку запишемо кінцевий результат, тоді за допомогою циклу ми проходимося по числах від 0 до 100 при тому будемо перевіряти їхню остачу від ділення, якщо вона 0 то це число парне і ми його додаємо.

Фактично затрачений час 25 хвилин.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/147

Завдання №2

```

1  #include <iostream>
2  #include <cmath>
3  #include <iomanip>
4
5  using namespace std;
6
7  const double N = 40;
8  const double PI = 3.141592;
9  const double A = PI / 5, B = (9 * PI) / 5, K = 10.0, E = 0.0001;
10
11 double f(double x) {
12     return -log(abs(2 * (sin(x / 2))));
13 }
14
15 double SN(double x) {
16     double sum = 1;
17     for (double i = 1; i <= N; i++) {
18         sum += cos(i * x) / i;
19     }
20     return sum;
21 }
22
23 double SE(double x) {
24     double sum = 1;
25     double prev;
26     double i = 1;
27     do {
28         prev = sum;
29         sum += cos(i * x) / i;
30         i++;
31     } while (abs(prev - sum) > E);
32     return sum;
33 }
34
35 int main() {
36     double step = (B - A) / K;
37     cout << fixed << setprecision(10);
38     for (double x = A; x <= B; x += step) {
39         cout << "X = " << x << "; SN = " << SN(x) << "; SE = " << SE(x) << "; Y = " << f(x) << endl;
40     }
41     return 0;
42 }

```

Рисунок 3.3. Код до програми №2

```

X = 0.6283184000; SN = 1.4920901479; SE = 1.4807879129; Y = 0.4812120262
X = 1.1309731200; SN = 0.9532048987; SE = 0.9296025582; Y = -0.0692026813
X = 1.6336278400; SN = 0.6199982328; SE = 0.6222788911; Y = -0.3770224380
X = 2.1362825600; SN = 0.4249062908; SE = 0.4393729702; Y = -0.5611079005
X = 2.6389372800; SN = 0.3395334484; SE = 0.3386194753; Y = -0.6612261762
X = 3.1415920000; SN = 0.3191966182; SE = 0.3069028159; Y = -0.6931471806
X = 3.6442467200; SN = 0.3395339541; SE = 0.3385275150; Y = -0.6612263440
X = 4.1469014400; SN = 0.4249060250; SE = 0.4393719811; Y = -0.5611082598
X = 4.6495561600; SN = 0.6199967383; SE = 0.6237776983; Y = -0.3770230518
X = 5.1522108800; SN = 0.9532035654; SE = 0.9295251906; Y = -0.0692037112
X = 5.6548656000; SN = 1.4920901478; SE = 1.4806808167; Y = 0.4812100147

```

Рисунок 3.4. Приклад виконання програми №2

Спочатку ми ініціалізуємо константи типу double, далі створюємо три функції які повертають значення f (для обчислення точного значення функції), SN(для

обчислення функції із заданою точністю n), SE (для обчислення функції із точністю епсілон).

Фактично затрачений час 2 години.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/147

Завдання №3

```
1  #include <iostream>
2  #include <cstdarg>
3  #include <iomanip>
4
5  using namespace std;
6
7  float mult (int count, float first, ...){
8      va_list arguments;
9      va_start (arguments, first);
10
11     float res = first;
12
13     for (int i = 1; i < count; i++){
14         double next = va_arg(arguments, double);
15         res *= static_cast<float>(next);
16     }
17     va_end(arguments);
18
19     return res;
20
21 }
22
23 int main(){
24     cout << fixed << setprecision(2);
25     float case1 = mult(3, 8.f, 5.f, 4.f);
26     cout << case1 << endl;
27     float case2 = mult(7, 45.f, 0.8, 34.f, 1.23, 52.f, 3.f, 5.f);
28     cout << case2 << endl;
29     float case3 = mult(11, 4.5, 34.f, 2.45, 76.f, 3.f, 6.f, 90.f, 5.f, 43.f, 1.f, 16.f);
30     cout << case3 << endl;
31
32     return 0;
33 }
```

Рисунок 3.5. Код до програми №3

```
160.00
1174305.62
158761271296.00
```

Рисунок 3.6. Приклад виконання програми №3

Спочатку ми створюємо функцію `mult` зі змінною кількістю параметрів типу `float`, яка буде їх додавати. У функції за допомогою `va_list arguments` ми створили список аргументів далі `va_start(arguments, first)` ініціалізували доступ до додаткових аргументів. У циклі за допомогою команди `va_arg(arguments, double)` отримали наступний елемент зі списку ну і `va_end(arguments)` завершили роботу зі списком. Відповідно у головній функції ми спочатку встановили скільки хочемо цифр після

коми для кожного результату, потім викликали функцію `mult` де вказали спочатку число наших параметрів і написали, які саме параметри потрібно додати.

Фактично затрачений час 2 години.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/147

Завдання №4

```
1  #include <iostream>
2  #include <complex>
3
4  using namespace std;
5
6  double addition_real_numbers(double r1, double r2){
7      return r1 + r2;
8  }
9
10 double addition_real_numbers(double r1, double r2, double r3){
11     return r1 + r2 + r3;
12 }
13
14 double addition_real_numbers(double r1, double r2, double r3, double r4){
15     return r1 + r2 + r3 + r4;
16 }
17
18 complex<int> addition_complex_numbers(complex<int> c1, complex<int> c2){
19     return c1 + c2;
20 }
21
22 complex<int> addition_complex_numbers(complex<int> c1, complex<int> c2, complex<int> c3){
23     return c1 + c2 + c3;
24 }
25
26 complex<int> addition_complex_numbers(complex<int> c1, complex<int> c2, complex<int> c3, complex<int> c4){
27     return c1 + c2 + c3 + c4;
28 }
29 int main(){
30
31     //add real numbers
32
33     cout << addition_real_numbers(8, 2.3) << endl;
34     cout << addition_real_numbers(7, 2.3, 4.3) << endl;
35     cout << addition_real_numbers(3, 2.5, 12.7, 3.8) << endl;
36
37     //add complex numbers
38
39     cout << addition_complex_numbers(3, 5) << endl;
40     cout << addition_complex_numbers((3, 7), (6, 2), (5, 4)) << endl;
41     cout << addition_complex_numbers((8, 6), (8, 10), (9, 1), (10, 3)) << endl;
42     return 0;
43 }
```

Рисунок 3.7. Код до програми №4

```
10.3
13.6
22
(8,0)
(13,0)
(20,0)
```

Рисунок 3.8. Приклад виконання програми №4

Спочатку створюємо 3 перевантажені функції `addition_real_numbers`, які будуть додавати три цілі числа, потім 3 перевантажені функції `addition_complex_numbers`, які відповідно будуть додавати комплексні числа. У головній функції ми будемо викликати тричі одну функцію і тричі другу, щоб додати різну кількість чисел.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/147

Завдання №5

```

1  #include <iostream>
2  #include <vector>
3  #include <string>
4
5  using namespace std;
6
7  vector<string> book_list ({
8      "1. 1984",
9      "2. Harry Potter and the Goblet of Fire",
10     "3. Footprints on the road",
11     "4. The Catcher in the Rye",
12     "5. Wuthering Heights",
13     "6. Fight Club",
14     "7. Fahrenheit 451",
15 });
16
17 vector<bool> available_books({ false, true, false, true, false, true, false });
18
19 int main() {
20     int option;
21     char choice;
22     int choosen_book;
23
24     do {
25         while (true) {
26             MainMenu:
27             cout << "1. Enumerate all books\n";
28             cout << "2. Allow to take book if it is available\n";
29             cout << "3. Allow to return the book\n";
30             cout << "4. Leave the library\n";
31             cout << "Choose an option: ";
32             cin >> option;
33
34             switch (option) {
35                 case 1:
36                     cout << "Here is your list of books:" << endl;
37                     for (const auto& book : book_list) {
38                         cout << book << endl;
39                     }
40                     cout << "-----" << endl;
41                     break;
42
43                 case 2:
44                     cout << "Available books:" << endl;
45                     for (int i = 0; i < book_list.size(); i++) {
46                         if (available_books[i]) {
47                             cout << book_list[i] << endl;
48                         }
49                     }
50
51                     while (true) {
52                         cout << "Enter the number of the book you want to take: ";
53                         cin >> choosen_book;
54
55                         // Check index validity
56                         if (choosen_book < 1 || choosen_book > book_list.size()) {
57                             cout << "Invalid number. Please choose a valid book number." << endl;
58                             continue;
59                         }
60
61                         if (available_books[choosen_book - 1]) {
62                             available_books[choosen_book - 1] = false;
63                             cout << "Thank you for choosing our library!" << endl;
64                             break;
65                         } else {
66                             cout << "This book is currently borrowed. Please choose another book." << endl;
67                         }
68                     }
69                     cout << "-----" << endl;
70                     break;
71
72                 case 3:
73                     int index;
74                     cout << "Enter the number of the book you want to return: ";
75                     cin >> index;
76
77                     // Check index validity
78                     if (index < 1 || index > book_list.size()) {
79                         cout << "We don't have this book, please enter a correct number." << endl;
80                     } else if (available_books[index - 1]) {
81                         cout << "This book is already in the library!" << endl;
82                     } else {
83                         available_books[index - 1] = true;
84                         cout << "Thank you for returning the book!" << endl;
85                     }
86                     cout << "-----" << endl;
87                     break;
88
89                 case 4:
90                     cout << "See you next time in our library!" << endl;
91                     return 0;
92
93                 default:
94                     cout << "You chose the wrong option, please try again." << endl;
95                     goto MainMenu;
96             }
97             break;
98         }
99
100        cout << "Would you like to perform another operation? (y/n): ";
101        cin >> choice;
102
103    } while (choice == 'y' || choice == 'Y');
104
105    cout << "Goodbye!" << endl;
106    return 0;
107 }
108

```

Рисунок 3.9. Код до програми №5

```

1. Enumerate all books
2. Allow to take book if it is available
3. Allow to return the book
4. Leave the library
Choose an option: 2
Available books:
2. Harry Potter and the Goblet of Fire
4. The Catcher in the Rye
6. Fight Club
Enter the number of the book you want to take: 3
This book is currently borrowed. Please choose another book.
Enter the number of the book you want to take: 2
Thank you for choosing our library!
-----
Would you like to perform another operation? (y/n): n
Goodbye!

```

Рисунок 3.10. Приклад виконання програми №5

Програма №5 це онлайн бібліотека. Користувачу на екран виводимо можливі дії і відносно того, що користувач уведе програма буде виводити різні результати на екран.

Фактично затрачений час 5 години.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/147

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; // Кількість компаній
10     vector<pair<int, int>> lengths(n); // Вектор пар для довжин і індексів
11
12     // Зчитування довжин офісів
13     for (int i = 0; i < n; i++) {
14         cin >> lengths[i].first; // Довжина офісу
15         lengths[i].second = i + 1; // Зберігаємо індекс (1-based)
16     }
17
18     // Сортуювання пар за довжинами
19     sort(lengths.begin(), lengths.end());
20
21     // Виведення порядку індексів
22     for (int i = 0; i < n; i++) {
23         cout << lengths[i].second << " "; // Виводимо індекси
24     }
25     cout << endl;
26
27     return 0;
28 }

```

Рисунок 3.11. Код до програми №6

```

5
3 5 9 3 5
1 4 2 5 3

```

Рисунок 3.12. Приклад виконання програми №6

Програма приймає від користувача число, тобто кількість офісів і записує вектор пар для довжин і індексів. Далі користувач заповнює вектор і ми зберігаємо індекси. Потім сортуємо і виводимо порядок індексів (компаній), які треба розташувати в оптимальному порядку, щоб відстань була найменшою.

Фактично затрачений час 2 години.

Посилання на файл у пул реквесті

4. Робота з командою:

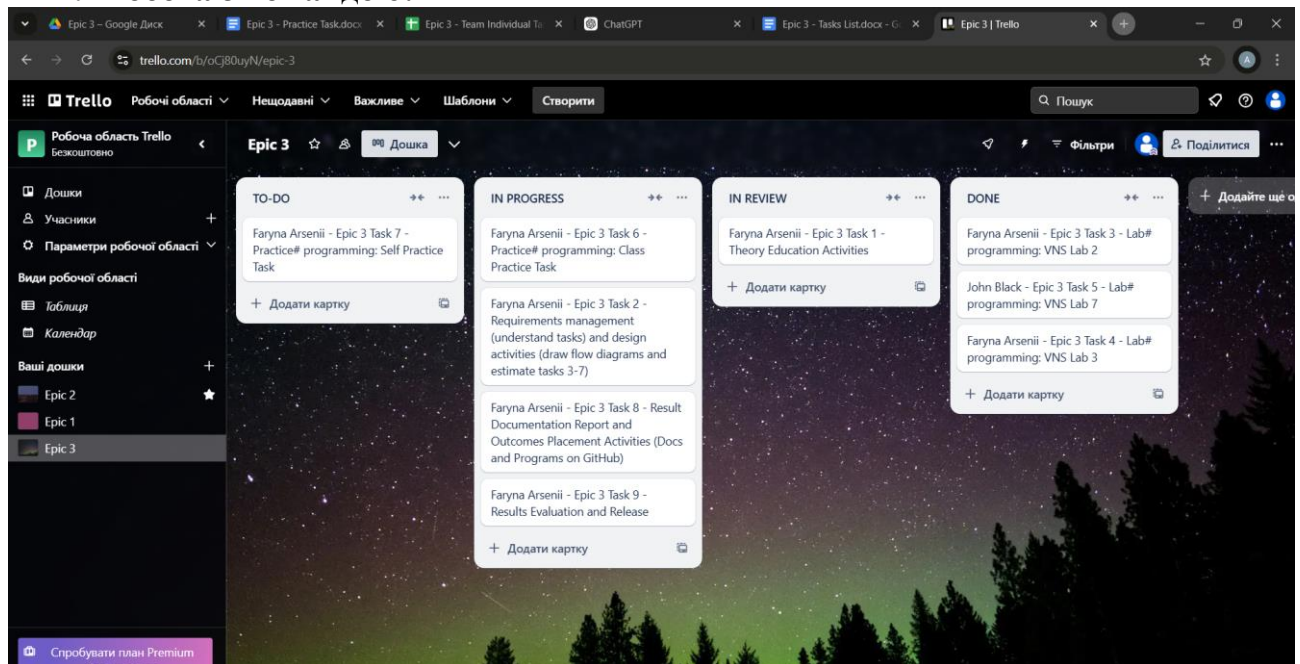


Рисунок 4.1. Командна дошка в Trello

Висновок: У межах практичних та лабораторних робіт блоку №3 я вивчив низку нових понять, таких як: функція, перевантажена функція, функція зі змінною кількістю параметрів (еліпсис), цикли, вкладені цикли, простір імен, завершення виконання циклів, рекурсія. Деякі з них довелося застосувати на практиці, що дало краще розуміння як це все працює на ділі. Для кращого розуміння як працює та чи інша програма я створив діаграми в Draw.io, тому також покращив роботу з цим редактором. Також створив чергову дошку в Trello для роботи в команді.