

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

Левченко Денис

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

16) Знайти суму ряду з точністю $\varepsilon=0.0001$, загальний член якого

$$a_n = \lg(n!)e^{-n\sqrt{n}}$$

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:

- для заданого n ;
- для заданої точності ε ($\varepsilon=0.0001$).

Для порівняння знайти точне значення функції.

16	$y = \frac{\pi^2}{8} - \frac{\pi}{4} x $	$\frac{\pi}{5} \leq x \leq \pi$	40	$S = \cos x + \frac{\cos 3x}{3^2} + \dots + \frac{\cos(2n-1)x}{(2n-1)^2}$
----	--	---------------------------------	----	---

Завдання №3

VNS LAB 7 – TASK 1

Написати функцію `days` зі змінною кількістю параметрів, яка знаходить кількість днів, що пройшли між двома датами (параметрами функції є дати у форматі «дд.мм.рр»). Написати викликаючу функцію `main`, що звертається до функції `days` не менше трьох разів з кількістю параметрів 3, 5, 8.

Завдання №4

VNS LAB 7 – TASK 2

Написати перевантажені функції й основну програму, що їх викликає.

2.

- для перетворення годин і хвилин у хвилини;
- для перетворення хвилин у години та хвилини.

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) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.
3. for: список усіх книг за допомогою циклу.
4. for each: перевірити наявність кожної книги.
5. goto: якщо користувач вводить неправильний вибір, використовуйте goto, щоб перенаправити його до головного меню.

Завдання №6

Self Practice Work Algotester

У вашого персонажа є N хітпойнтів та M мани.

Персонаж 3 рази використає закляття, кожне з яких може використати хітпойнти та ману одночасно.

Якщо якесь закляття забирає і хітпойнти і ману - ваш персонаж програє, отже для виграшу треба використовувати при одному заклинанні **АБО** хітпойнти, **АБО** ману.

Якщо в кінці персонаж буде мати додатню кількість хітпойнтів та мани ($N, M > 0$) - він виграє, в іншому випадку програє.

Ваше завдання у випадку виграшу персонажа вивести **YES**, вивести **NO** у іншому випадку.

Вхідні дані

2 цілих числа NN та MM - хітпойнти та мана персонажа

3 рядки по 2 цілих числа, hi та mi - кількість хітпойнтів та мани, які ваш персонаж потратить за хід на i заклинання

Вихідні дані

YES - якщо ваш персонаж виграє

NO - у всіх інших випадках

2. Дизайн та планувальна оцінка часу виконання завдань:

Програма №1

- Блок-схема

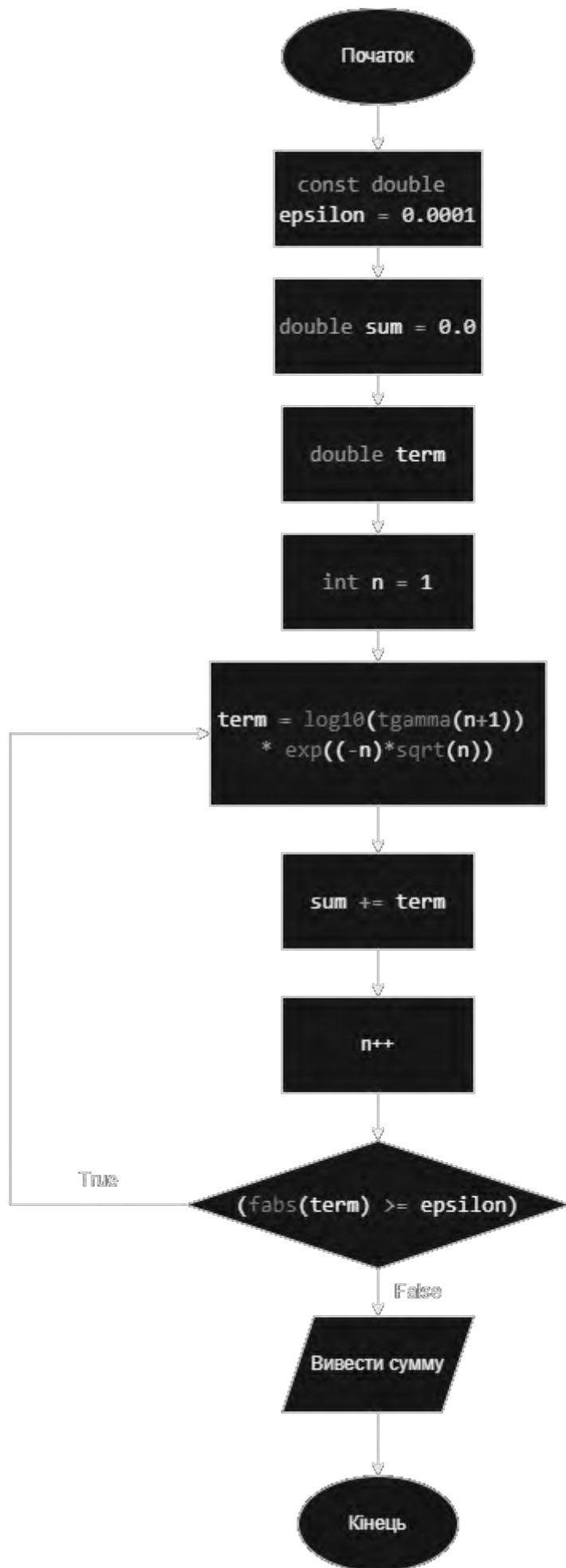


Рисунок 2.1. Блок-схема до програми 1

Програма №2

- Блок-схема
- Плановий час на реалізацію 2.5 години.

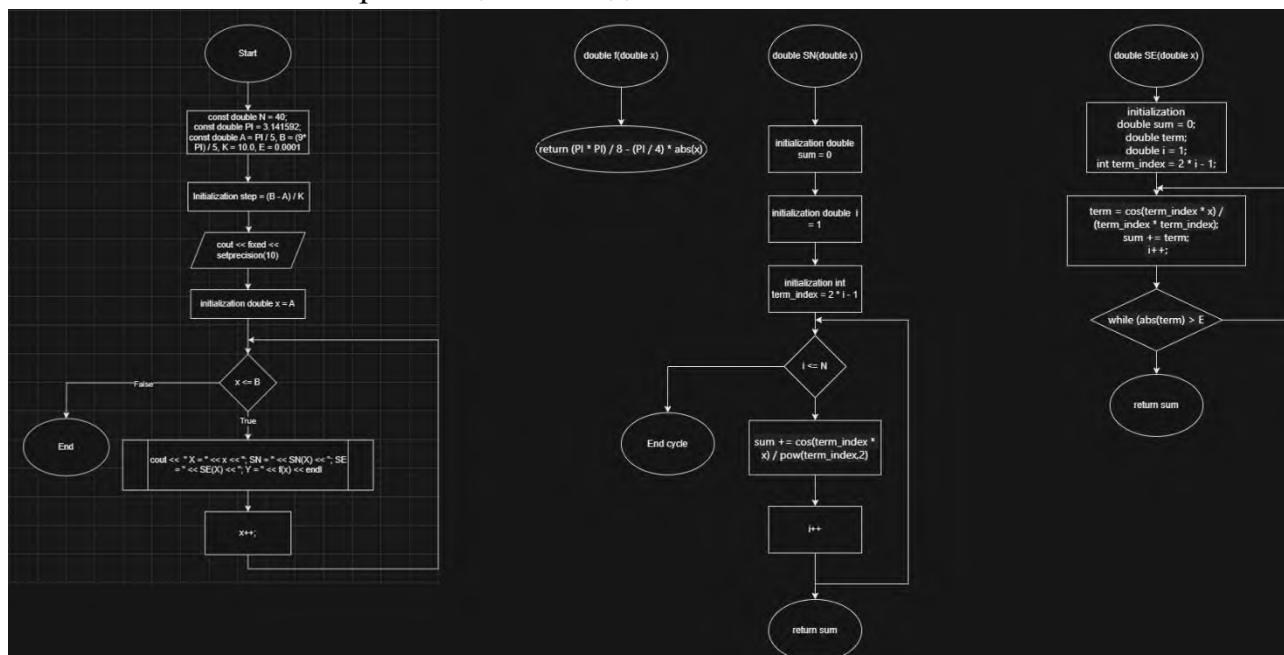


Рисунок 2.2. Блок-схема до програми 2

Програма №3

- Блок-схема
- Плановий час на реалізацію 1.5 години.

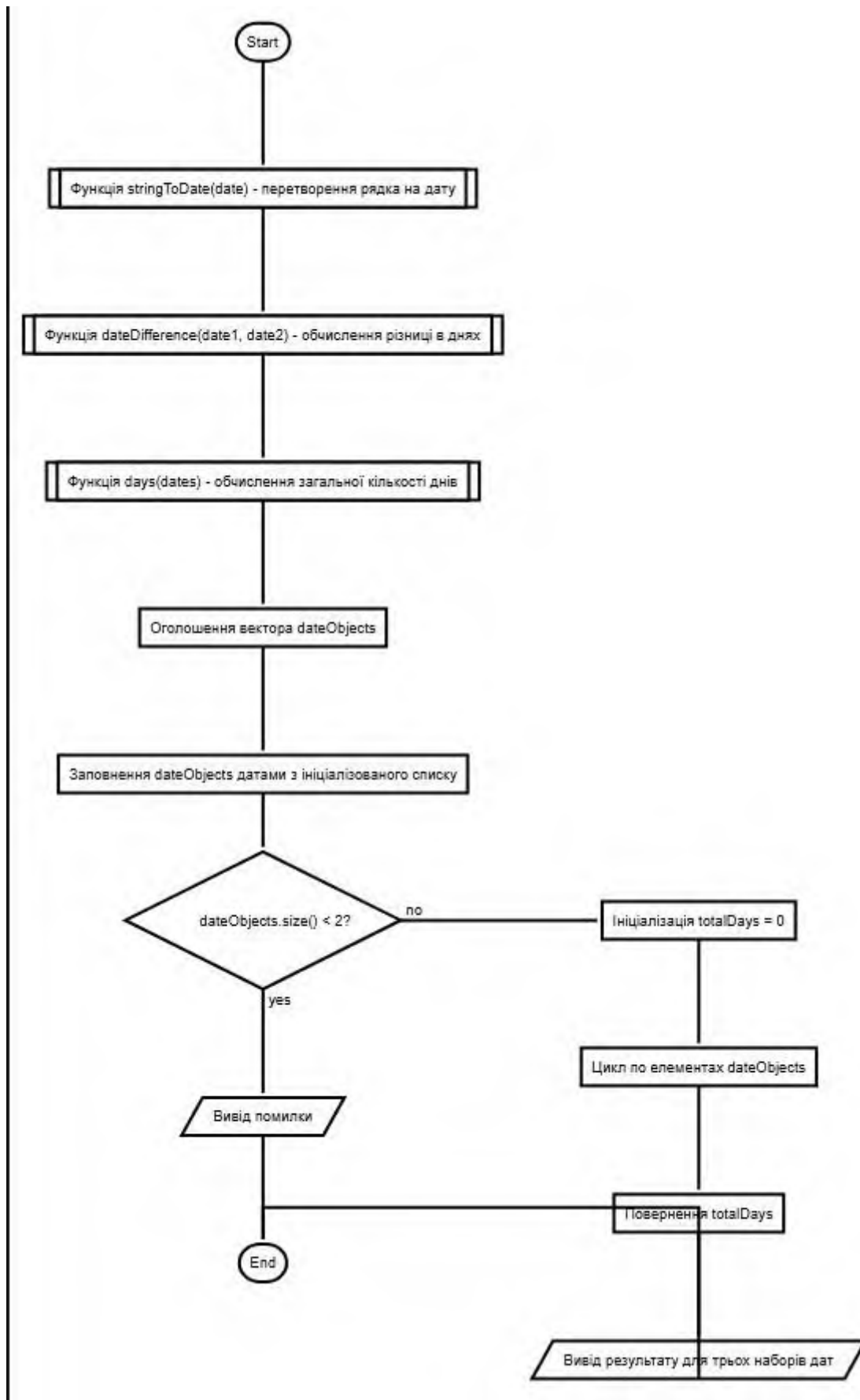


Рисунок 2.3. Блок-схема до програми 3

Програма №4

- Блок-схема
- Плановий час на реалізацію 2 години.

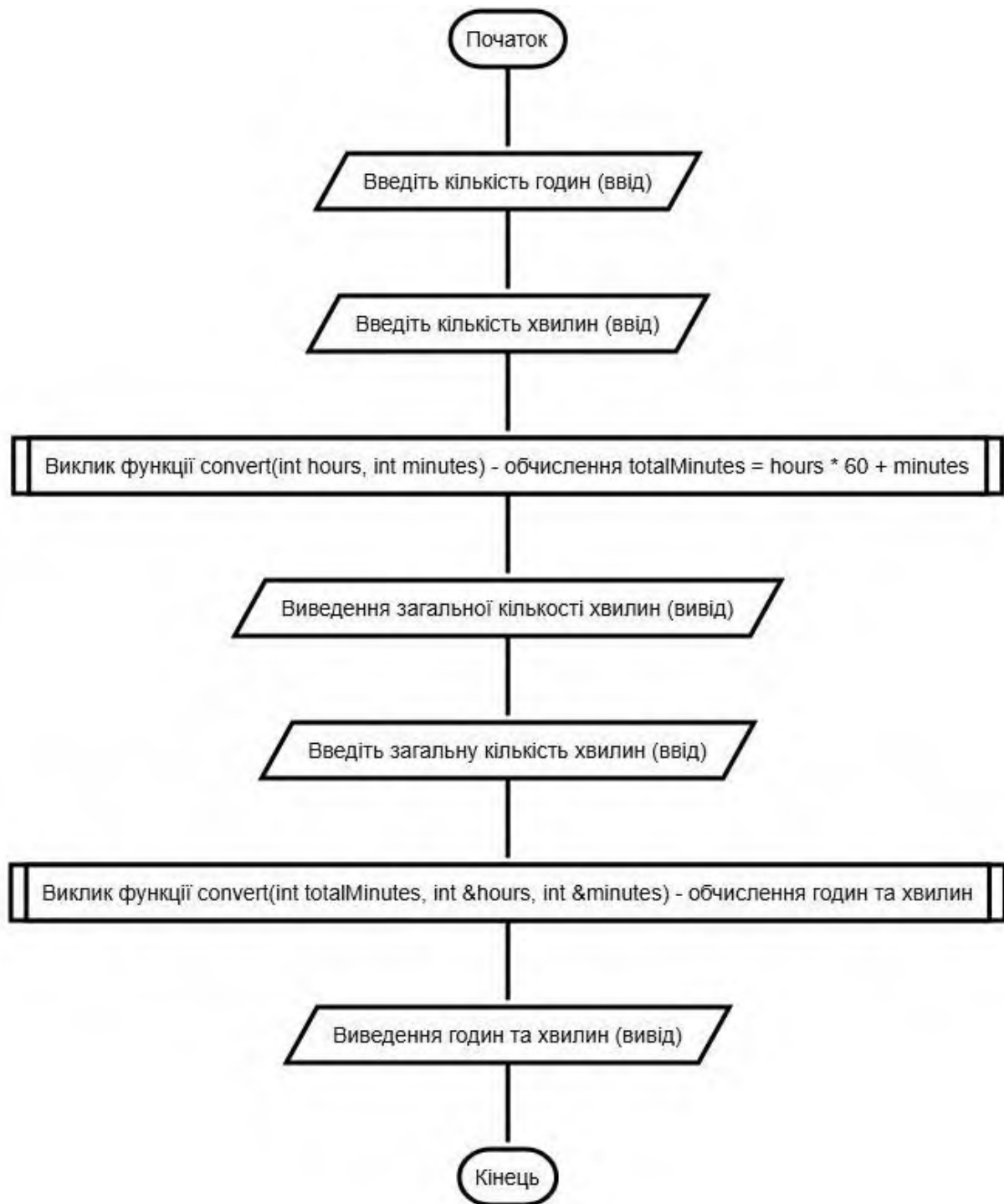


Рисунок 2.4. Блок-схема до програми 4

Програма №5

- Блок-схема

- Плановий час на реалізацію 1 день

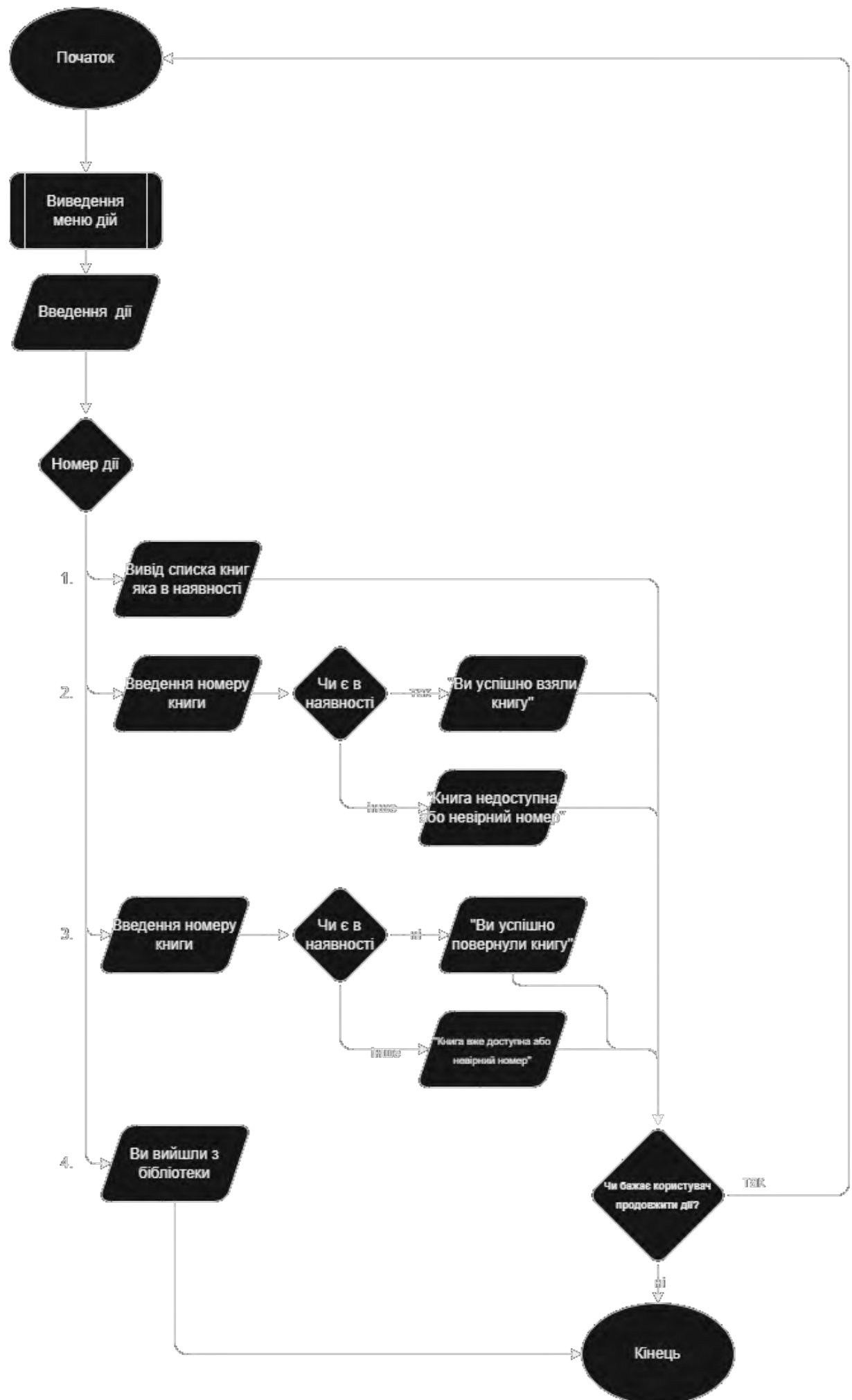


Рисунок 2.5. Блок-схема до програми №5

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

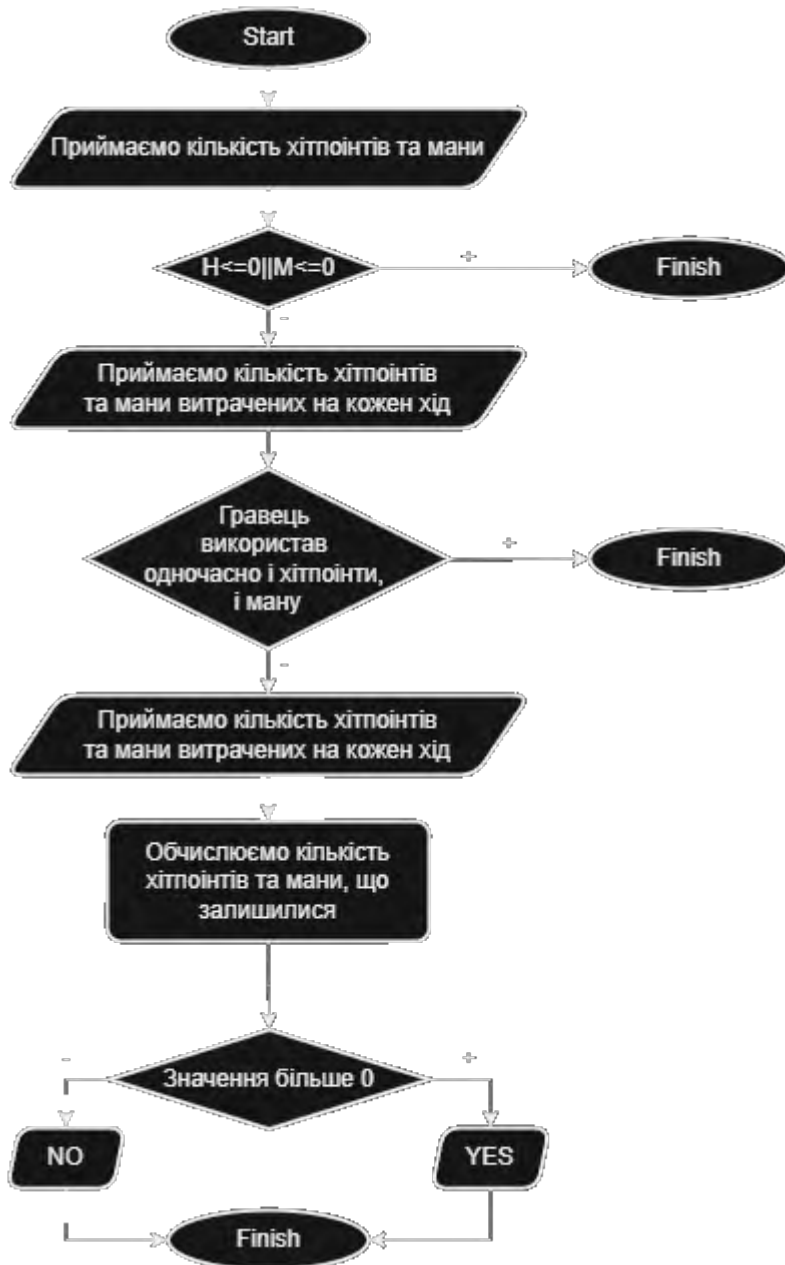


Рисунок 2.7. Блок-схема до програми №6

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

Завдання №1

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main(){
7      const double epsilon = 0.0001;
8      double sum = 0.0;
9      double term;
10     int n = 1;
11     do {
12         term = log10(tgamma(n+1)) * exp((-n)*sqrt(n));
13         sum += term;
14         n++;
15     } while (fabs(term) >= epsilon);
16     cout << sum << endl;
17 }

```

Рисунок 3.1. Код до програми №1

```

0
PS C:\Users\samykon>

```

Рисунок 3.2. Приклад виконання програми №1

Фактично затрачений час 25 хвилин.

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 = 3.141592653589793;
9  const double A = PI / 5, B = PI;
10 const double K = 10.0;
11 const double E = 0.0001;
12
13 double f(double x) {
14     return (PI * PI) / 8 - (PI / 4) * abs(x);
15 }
16
17 double SN(double x) {
18     double sum = 0;
19     for (double i = 1; i <= N; i++) {
20         int term_index = 2 * i - 1;
21         sum += cos(term_index * x) / pow(term_index, 2);
22     }
23     return sum;
24 }
25
26 double SE(double x) {
27     double sum = 0;
28     double term;
29     double i = 1;
30     do {
31         int term_index = 2 * i - 1;
32         term = cos(term_index * x) / (term_index * term_index);
33         sum += term;
34         i++;
35     } while (abs(term) > E);
36     return sum;
37 }
38
39 int main() {
40     double step = (B - A) / K;
41     cout << fixed << setprecision(10);
42     for (double x = A; x <= B; x += step) {
43         cout << "X = " << x
44             << "; SN = " << SN(x)
45             << "; SE = " << SE(x)
46             << "; Y = " << f(x) << endl;
47     }
48     return 0;
49 }
50

```

Рисунок 3.3. Код до програми №2

```

X = 0.6283185307; SN = 0.7402157797; SE = 0.7399771763; Y = 0.7402203301
X = 0.8796459430; SN = 0.5429239201; SE = 0.5425669627; Y = 0.5428282421
X = 1.1309733553; SN = 0.3454876908; SE = 0.3452936370; Y = 0.3454361540
X = 1.3823007676; SN = 0.1479976471; SE = 0.1481039595; Y = 0.1480440660
X = 1.6336281799; SN = -0.0494223971; SE = -0.0493213679; Y = -0.0493480220
X = 1.8849555922; SN = -0.2467394439; SE = -0.2467385333; Y = -0.2467401100
X = 2.1362830044; SN = -0.4440438192; SE = -0.4443584340; Y = -0.4441321980
X = 2.3876104167; SN = -0.6414597554; SE = -0.6422774102; Y = -0.6415242861
X = 2.6389378290; SN = -0.8390172784; SE = -0.8385217664; Y = -0.8389163741
X = 2.8902652413; SN = -1.0365938546; SE = -1.0381733265; Y = -1.0363084621
X = 3.1415926536; SN = -1.2274508756; SE = -1.2287987464; Y = -1.2337005501
PS C:\Users\samykon>

```

Рисунок 3.4. Приклад виконання програми №2

Фактично затрачений час 2 години.

Завдання №3

```

1  #include <iostream>
2  #include <sstream>
3  #include <vector>
4  #include <iomanip>
5
6  using namespace std;
7
8
9  tm stringToDate(const string &date) {
10     tm tm = {};
11     istringstream ss(date);
12     ss >> get_time(&tm, "%d.%m.%y");
13     return tm;
14 }
15
16
17 int dateDifference(const tm &date1, const tm &date2) {
18     time_t time1 = mktime(const_cast<tm*>(&date1));
19     time_t time2 = mktime(const_cast<tm*>(&date2));
20     return difftime(time2, time1) / (60 * 60 * 24);
21 }
22
23
24 int days(initializer_list<string> dates) {
25     vector<tm> dateObjects;
26     for (const auto &date : dates) {
27         dateObjects.push_back(stringToDate(date));
28     }
29     if (dateObjects.size() < 2) {
30         cerr << "Помилка: має бути передано принаймні дві дати." << endl;
31         return 0;
32     }
33     int totalDays = 0;
34     for (size_t i = 0; i < dateObjects.size() - 1; ++i) {
35         totalDays += dateDifference(dateObjects[i], dateObjects[i + 1]);
36     }
37     return totalDays;
38 }
39
40
41 int main() {
42     cout << "Кількість днів між датами (3 параметри): " << days({"01.01.23", "10.05.24", "6.01.25"}) << endl;
43     cout << "Кількість днів між датами (5 параметрів): " << days({"01.01.21", "02.02.22", "03.03.23", "04.04.24", "05.05.25"}) << endl;
44     cout << "Кількість днів між датами (8 параметрів): " << days({"01.03.24", "06.03.24", "11.04.24", "15.04.24", "01.05.24", "15.05.24", "01.06.24", "15.06.24"}) << endl;
45
46     return 0;
47 }
48

```

Рисунок 3.5. Код до програми №3

```

Кількість днів між датами (3 параметри): 736
Кількість днів між датами (5 параметрів): 1585
Кількість днів між датами (8 параметрів): 106
PS C:\Users\gazzie>

```

Рисунок 3.6. Приклад виконання програми №3

Фактично затрачений час 2 години.

Завдання №4

```
1  #include <iostream>
2  using namespace std;
3
4  int convert(int hours, int minutes) {
5      return hours * 60 + minutes;
6  }
7
8  void convert(int totalMinutes, int &hours, int &minutes) {
9      hours = totalMinutes / 60;
10     minutes = totalMinutes % 60;
11 }
12
13 int main() {
14     int hours, minutes, totalMinutes;
15
16     cout << "Введіть кількість годин: ";
17     cin >> hours;
18     cout << "Введіть кількість хвилин: ";
19     cin >> minutes;
20     totalMinutes = convert(hours, minutes);
21     cout << "Загальна кількість хвилин: " << totalMinutes << endl;
22
23     cout << "\nВведіть загальну кількість хвилин: ";
24     cin >> totalMinutes;
25     convert(totalMinutes, hours, minutes);
26     cout << "Години: " << hours << ", хвилини: " << minutes << endl;
27
28     return 0;
29 }
```

Рисунок 3.7. Код до програми №4

```
Введіть кількість годин: 7
Введіть кількість хвилин: 80
Загальна кількість хвилин: 500

Введіть загальну кількість хвилин: 1820
Години: 30, хвилини: 20
PS C:\Users\samykon>
```

Рисунок 3.8. Приклад виконання програми №4

Завдання №5

```

1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main() {
7     const int num_books = 5;
8     string books[num_books] = {"Harry Potter", "Lord of the Rings", "The Little Prince", "One Hundred Years of Solitude", "Kobzar"};
9     bool availability[num_books] = {true, true, true, true, true};
10     int choice;
11     string user_choice;
12
13     main_menu:
14     while (true) {
15         cout << "\nГоловне меню:\n";
16         cout << "1. Переглянути всі книги\n";
17         cout << "2. Взяти книгу\n";
18         cout << "3. Повернути книгу\n";
19         cout << "4. Вийти\n";
20         cout << "Введіть свій вибір: ";
21         cin >> choice;
22
23         switch (choice) {
24             case 1: {
25                 cout << "\nСписок книг:\n";
26                 for (int i = 0; i < num_books; i++) {
27                     cout << i + 1 << ". " << books[i] << (availability[i] ? " (Доступна)" : " (Позичена)") << "\n";
28                 }
29                 break;
30             }
31             case 2: {
32                 cout << "\nВведіть номер книги, яку хочете взяти: ";
33                 int book_number;
34                 cin >> book_number;
35
36                 if (book_number < 1 || book_number > num_books) {
37                     cout << "Невірний номер книги. Спробуйте ще раз.\n";
38                     goto main_menu;
39                 }
40
41                 if (availability[book_number - 1]) {
42                     availability[book_number - 1] = false;
43                     cout << "Ви успішно взяли книгу: " << books[book_number - 1] << "\n";
44                 } else {
45                     cout << "Ця книга вже позичена.\n";
46                 }
47                 break;
48             }
49             case 3: {
50                 cout << "\nВведіть номер книги, яку хочете повернути: ";
51                 int book_number;
52                 cin >> book_number;
53
54                 if (book_number < 1 || book_number > num_books) {
55                     cout << "Невірний номер книги. Спробуйте ще раз.\n";
56                     goto main_menu;
57                 }
58
59                 if (availability[book_number - 1]) {
60                     availability[book_number - 1] = true;
61                     cout << "Ви успішно повернули книгу: " << books[book_number - 1] << "\n";
62                 } else {
63                     cout << "Ця книга вже доступна.\n";
64                 }
65                 break;
66             }
67             case 4: {
68                 cout << "Вихід з програми...\n";
69                 return 0;
70             }
71             default: {
72                 cout << "Невірний вибір. Спробуйте ще раз.\n";
73                 goto main_menu;
74             }
75         }
76
77         do {
78             cout << "\nБажаєте виконати іншу операцію? (y/n): ";
79             cin >> user_choice;
80             if (user_choice == "n" || user_choice == "N") {
81                 cout << "Вихід з програми...\n";
82                 return 0;
83             } else if (user_choice == "y" || user_choice == "Y") {
84                 goto main_menu;
85             } else {
86                 cout << "Невірний вибір. Спробуйте ще раз.\n";
87             }
88         } while (user_choice != "y" && user_choice != "n" && user_choice != "Y" && user_choice != "N");
89     }
90
91     return 0;
92 }

```

Рисунок 3.9. Код до програми №5


```
Головне меню:
1. Переглянути всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти
Введіть свій вибір: 1

Список книг:
1. Harry Potter (Доступна)
2. Lord of the Rings (Доступна)
3. The Little Prince (Доступна)
4. One Hundred Years of Solitude (Доступна)
5. Kobzar (Доступна)

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

Головне меню:
1. Переглянути всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти
Введіть свій вибір: 2

Введіть номер книги, яку хочете взяти: 5
Ви успішно взяли книгу: Kobzar

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

Головне меню:
1. Переглянути всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти
Введіть свій вибір: 4
Вихід з програми...
PS C:\Users\samykon> █
```

Рисунок 3.10. Приклад виконання програми №5

Фактично затрачений час 5 години.

Завдання №6

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      long H; long M;
6      long sumH; long sumM;
7      long resH; long resM;
8      long move1[2]; long move2[2]; long move3[2];
9      cin>>H >>M;
10     if (H<=0||M<=0){return 0;}
11     else
12         cin>>move1[0]>>move1[1];
13         cin>>move2[0]>>move2[1];
14         cin>>move3[0]>>move3[1];
15     if (move1[0]<0||move1[1]<0||move2[0]<0||move2[1]<0||move3[0]<0||move3[1]<0){
16         return 0;
17     }
18     else
19         if((move1[0]>0&&move1[1]>0)|| (move2[0]>0&&move2[1]>0)|| (move3[0]>0&&move3[1]>0)){
20             cout<<"NO";
21             return 0;}
22     else
23         sumH=move1[0]+move2[0]+move3[0];
24         resH=H-sumH;
25         sumM=move1[1]+move2[1]+move3[1];
26         resM=M-sumM;
27         if(resM>0&&resH>0)
28             cout<<"YES";
29         else cout<<"NO";
30
31     return 0;}

```

Рисунок 3.11. Код до програми №6

```

100 100
10 0
10 0
79 0
YES
PS C:\Users\samykon>

```

Рисунок 3.12. Приклад виконання програми №6

Фактично затрачений час 2 години.

4. Робота з командою:

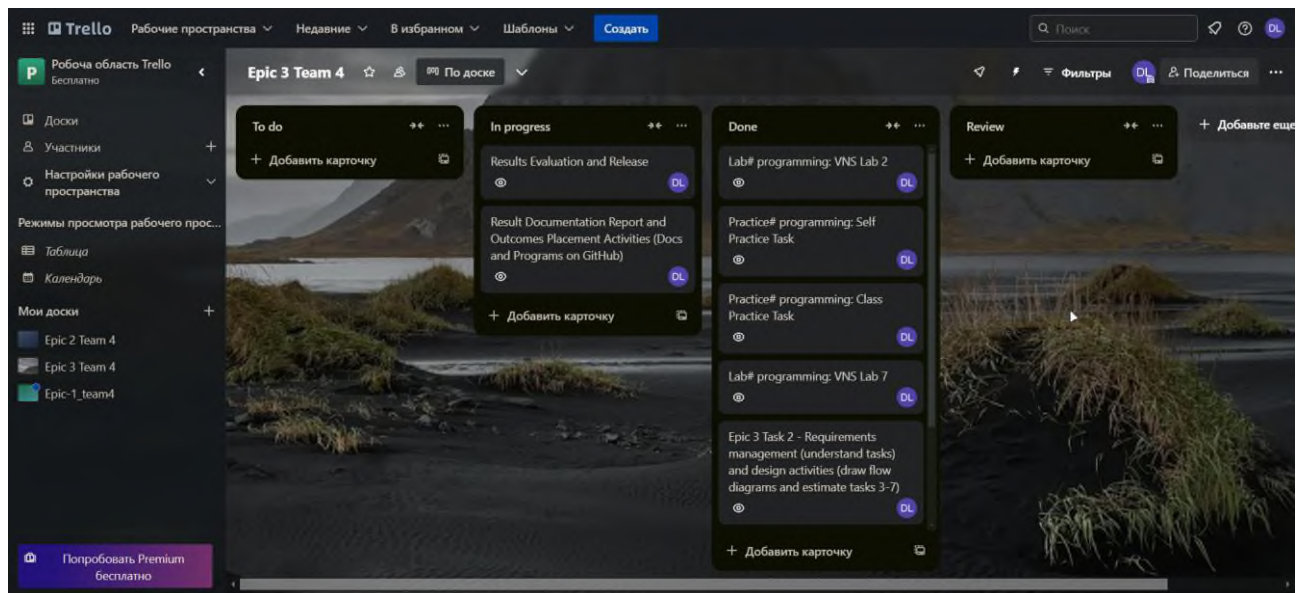


Рисунок 4.1. Командна дошка в Trello

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.