



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 ШІ-11

Єдинець Євген Русланович

Тема роботи:

Вивчення основ програмування на мові C++: цикли, вкладені цикли, завершення циклів, функції, простір імен, перевантаження функцій, функції зі змінною кількістю параметрів, рекурсія та вбудовані функції.

Мета роботи:

Ознайомитися з основами роботи з циклами та вкладеними циклами, вивчити способи завершення виконання циклів. Набути навичок створення та використання функцій, включаючи перевантаження функцій, роботу з просторами імен, функції зі змінною кількістю параметрів та рекурсивні функції. Дослідити можливості застосування вбудованих функцій мови C++.

Теоретичні відомості:

1. Поняття циклу

<https://www.youtube.com/watch?v=zBtcqNdiRf4&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=31>

2. Функції

<https://www.youtube.com/watch?v=G8P6SvdqU9s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=43>

3. Цикли та оператори в них (For, While, Do While)

<https://www.youtube.com/watch?v=LYIPTmN37SU&list=PL7vq4D0vOpQa9WaLe7btV01eixBUZ6-Ve&index=7>

4. Перевантаження функцій

<https://www.youtube.com/watch?v=3KJfisev6SI&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=48>

5. Простір імен

<https://www.youtube.com/watch?v=hcYgFCgeZzQ&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=106>

Виконання роботи:

1) Опрацювання завдання та вимог до програм та середовища:

Завдання 1

VNS Lab 2

Використовуючи оператор циклу, знайти суму елементів, зазначених у конкретному варіанті. Результат надрукувати, надавши відповідний заголовок.

Знайти суму 13 членів ряду, у якому:

$$a_n = \frac{\ln(n!)}{n^2}$$

Завдання 2

VNS Lab 3

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.

функція	діапазон зміни аргументу	n	сума
$y = \arctg x$	$0,1 \leq x \leq 1$	40	$S = x - \frac{x^3}{3} + \dots (-1)^n \frac{x^{2n+1}}{2n+1}$

Завдання 3

VNS Lab 7 - Task 1

Написати функцію (або макровизначення), що знаходить кут трикутника за його сторонами. Написати функцію `angles` с змінною кількістю параметрів, що знаходить кути n-кутника за заданими сторонами. Написати викликаючу функцію `main`, що звертається до функції `angle` не менше трьох разів з кількістю параметрів 3, 9, 11.

Завдання 4

VNS Lab 7 - Task 2

Написати перевантажені функції й основну програму, що їх викликає.

- а) для масиву цілих чисел знаходить кількість від'ємних елементів;
- б) для рядка знаходить кількість слів, що закінчуються і починаються на ту ж букву.

Завдання 5

Class Practice Work

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.

Завдання 6

Self Practice Work

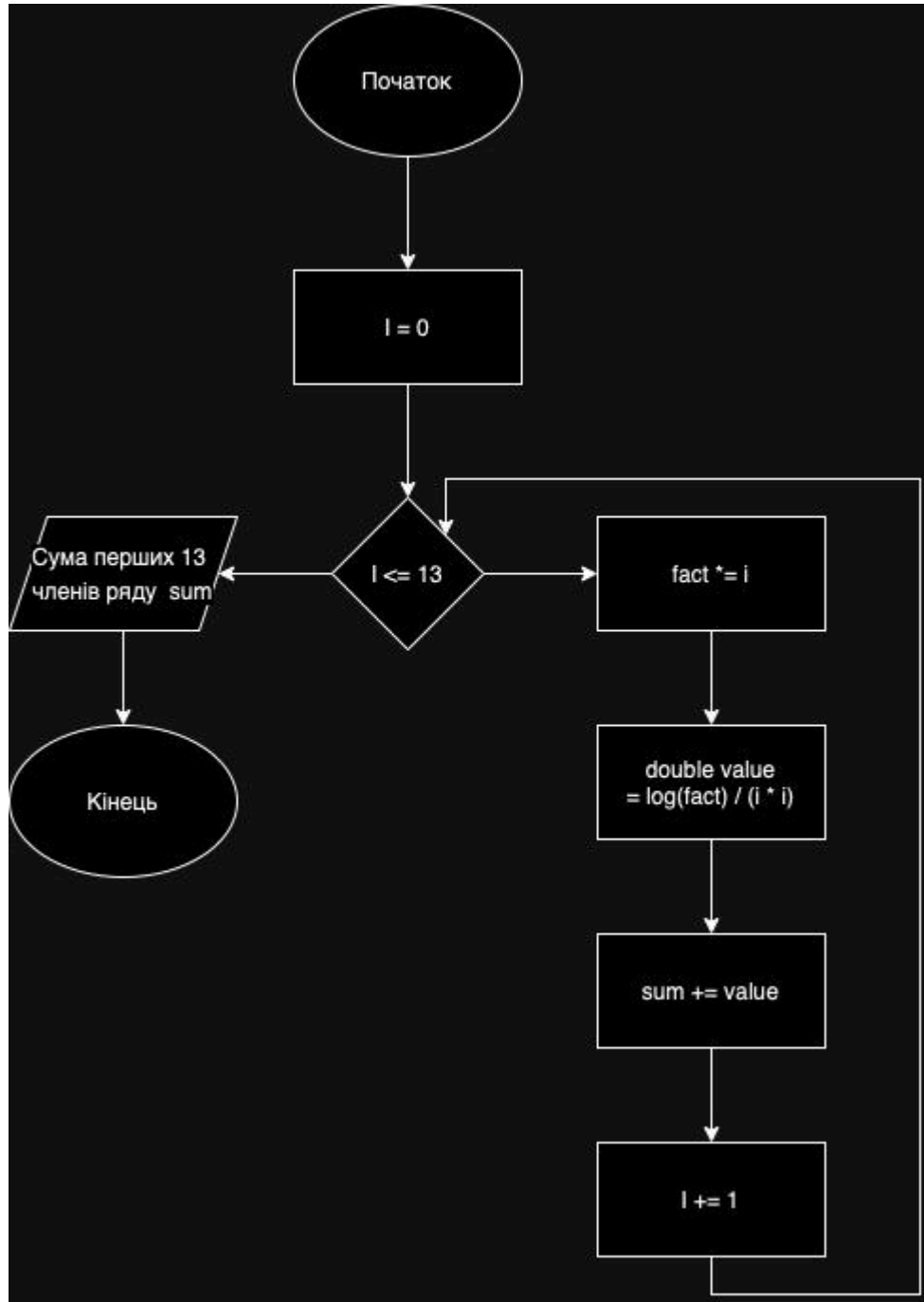
Замовлення від стоматологів (2106)

<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/71060#mySolutions>

2) Дизайн та планована оцінка часу виконання завдань:

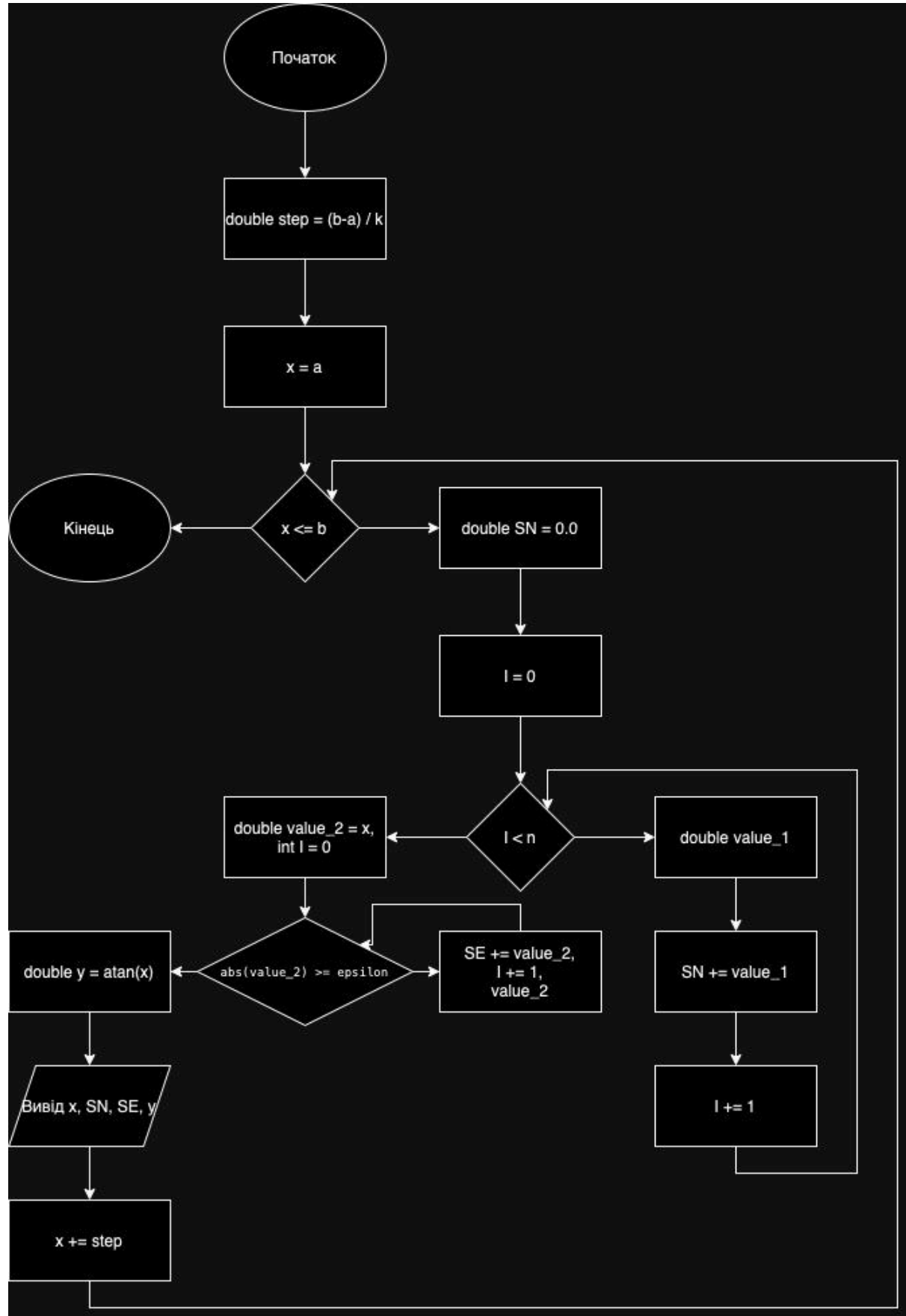
Завдання 1

Планований час виконання: 15- 30 хв.



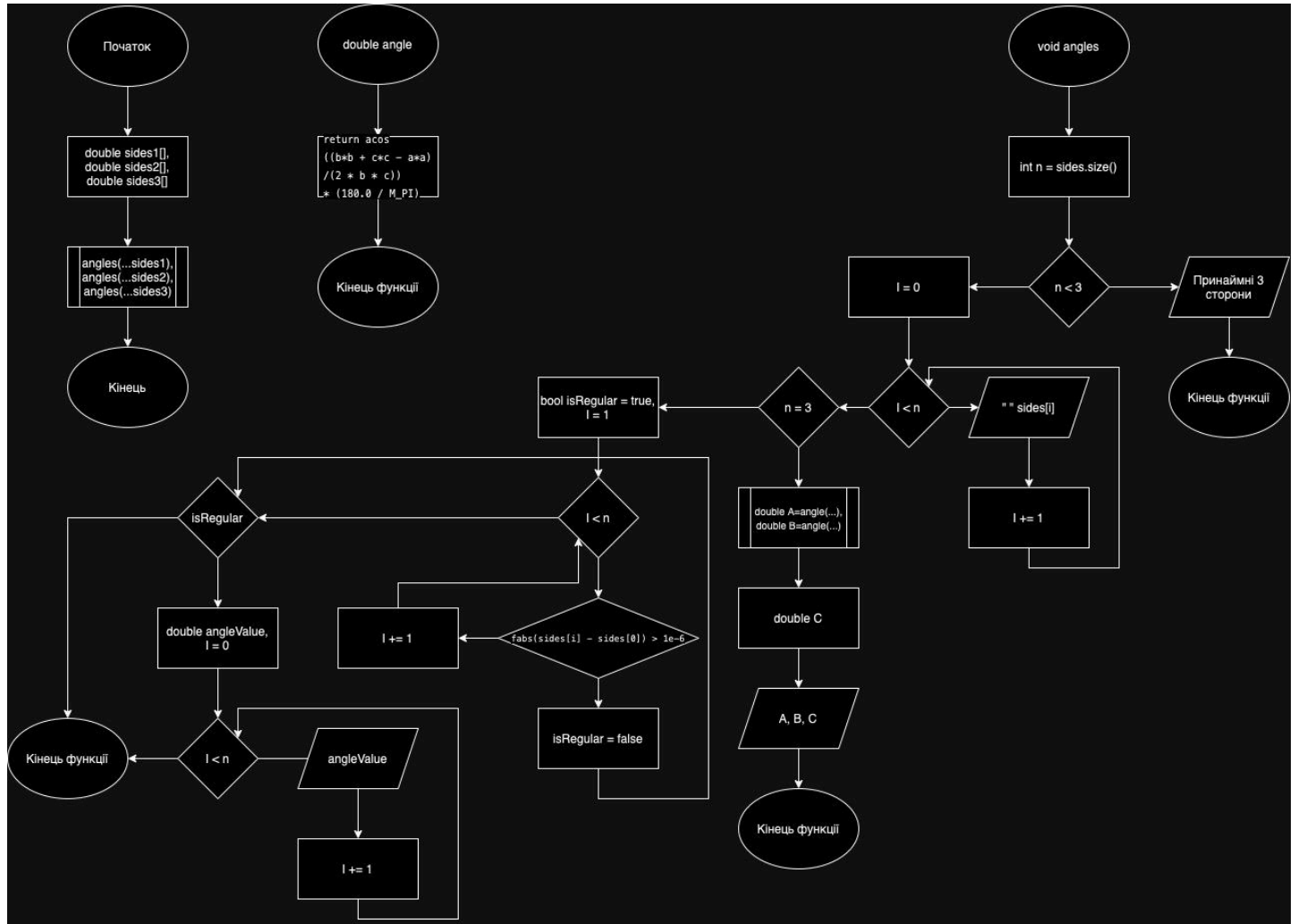
Завдання 2:

Планований час виконання: 30-45 хв.



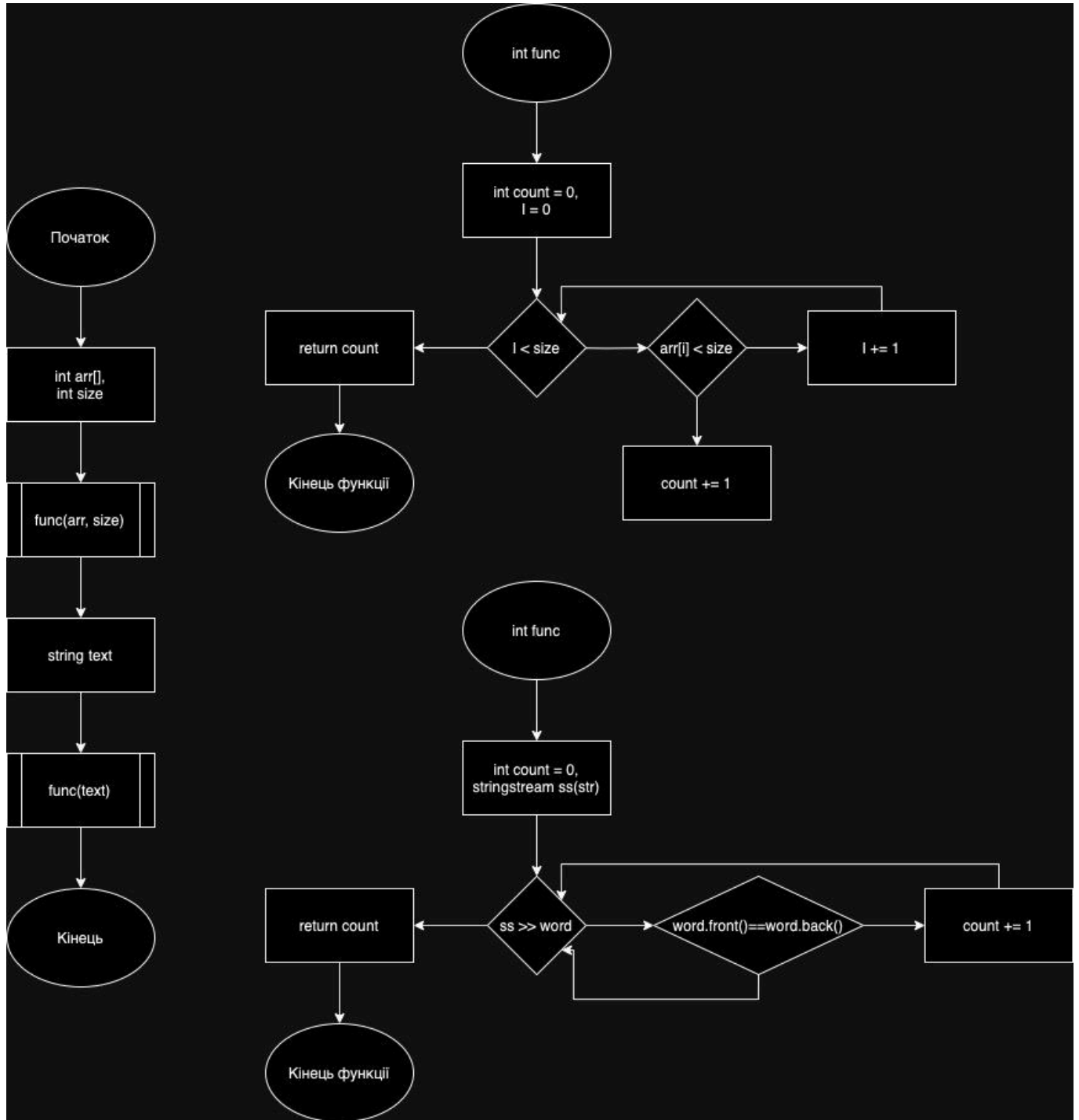
Завдання 3:

Планований час виконання: 40-60 хв.



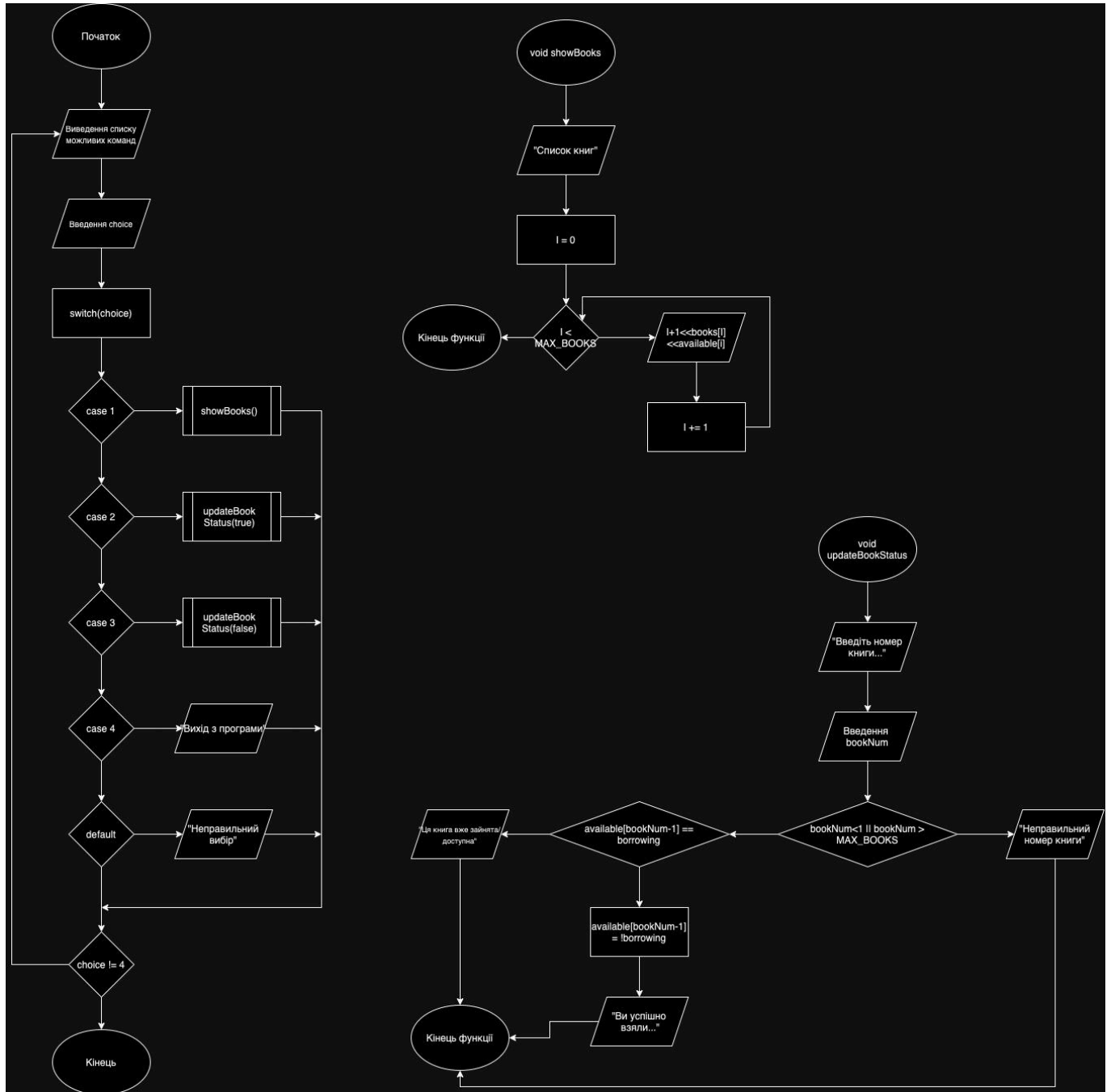
Завдання 4:

Планований час виконання: 30-45 хв.



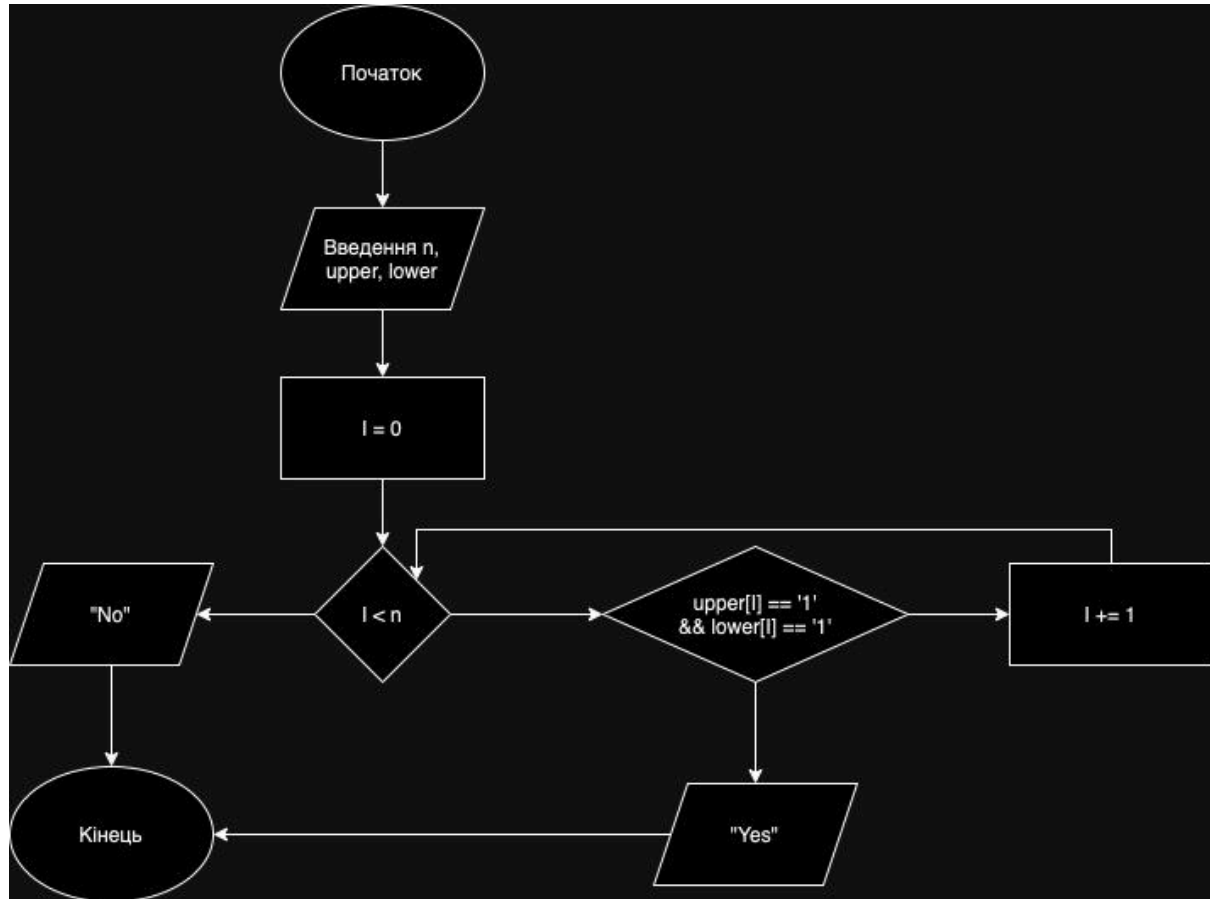
Завдання 5:

Планований час виконання: 45-60 хв.



Завдання 6:

Планований час виконання: 15-20 хв.



3) Код програм:

Завдання 1:

```

1  #include<iostream>
2  #include<math.h>
3
4  using namespace std;
5
6  int main(){
7      int n = 13;
8      double sum = 0.0;
9      double fact = 1.0;
10
11     for (int i = 1; i <= 13; ++i){
12         fact *= i;
13         double value = log(fact) / (i * i);
14         sum += value;
15     }
16
17     cout<< "Сума перших 13 членів ряду: " << sum;
18
19     return 0;
20 }

```

Завдання 2:

```
1  #include<iostream>
2  #include<iomanip>
3  #include<math.h>
4
5  using namespace std;
6
7  int main(){
8      double a = 0.1;
9      double b = 1.0;
10     int n = 40;
11     int k = 10;
12     double epsilon = 0.0001;
13     double step = (b - a) / k;
14
15     for (double x = a; x <= b; x += step){
16         double SN = 0.0;
17
18         for (int i = 0; i < n; ++i){
19             double value_1 = pow(-1, i) * pow(x, 2 * i + 1) / (2 * i + 1);
20             SN += value_1;
21         }
22
23         double SE = 0.0;
24         double value_2 = x;
25         int i = 0;
26
27         while (abs(value_2) >= epsilon){
28             SE += value_2;
29             i += 1;
30             value_2 = pow(-1, i) * pow(x, 2 * i + 1) / (2 * i + 1);
31         }
32
33         double y = atan(x);
34         cout<< "X = " << x << "    SN = " << SN << "    SE = " << SE << "    Y = " << y << endl;
35     }
36
37     return 0;
38 }
```

Завдання 3:

```

1  #include <iostream>
2  #include <math.h>
3  #include <vector>
4
5  using namespace std;
6
7  double angle(double a, double b, double c) {
8      return acos((b*b + c*c - a*a) / (2 * b * c)) * (180.0 / M_PI);
9  }
10
11 void angles(const vector<double>& sides) {
12     int n = sides.size();
13     if (n < 3) {
14         cout << "Необхідно принаймні 3 сторони для обчислення кутів" << endl;
15         return;
16     }
17
18     for (int i = 0; i < n; ++i) {
19         cout << " " << sides[i];
20     }
21     cout << endl;
22
23     if (n == 3) {
24         double A = angle(sides[0], sides[1], sides[2]);
25         double B = angle(sides[1], sides[0], sides[2]);
26         double C = 180.0 - A - B;
27         cout << "Кути трикутника: A = " << A << ", B = " << B << ", C = " << C << endl;
28         return;
29     }
30
31     bool isRegular = true;
32     for (int i = 1; i < n; ++i) {
33         if (fabs(sides[i] - sides[0]) > 1e-6) {
34             isRegular = false;
35             break;
36         }
37     }
38
39     if (isRegular) {
40         double angleValue = ((n - 2) * 180.0) / n;
41         cout << "Кути правильного " << n << "-кутника: ";
42         for (int i = 0; i < n; ++i) {
43             cout << angleValue << " ";
44         }
45         cout << endl;
46         return;
47     } else {
48         cout << "Обчислення кутів для нерівностороннього n-кутника ще не реалізовано" << endl;
49     }
50 }
51
52 int main() {
53     double sides1[] = {3.0, 4.0, 5.0};
54     double sides2[] = {3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0};
55     double sides3[] = {10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0};
56
57     angles(vector<double>(sides1, sides1 + 3));
58     angles(vector<double>(sides2, sides2 + 9));
59     angles(vector<double>(sides3, sides3 + 11));
60
61     return 0;
62 }

```

Завдання 4:

```

1  #include <iostream>
2  #include <sstream>
3
4  using namespace std;
5
6  int func(const int arr[], int size) {
7      int count = 0;
8      for (int i = 0; i < size; ++i) {
9          if (arr[i] < 0) {
10             count++;
11         }
12     }
13     return count;
14 }
15
16 int func(const string& str) {
17     int count = 0;
18     stringstream ss(str);
19     string word;
20
21     while (ss >> word) {
22         if (word.front() == word.back()) {
23             count++;
24         }
25     }
26     return count;
27 }
28
29 int main() {
30     int arr[] = {-5, 3, -2, 0, -7, 8};
31     int size = sizeof(arr) / sizeof(arr[0]);
32     cout << "Кількість від'ємних елементів: " << func(arr, size) << endl;
33
34     string text = "john level civic world";
35     cout << "Кількість слів, що починаються та закінчуються на ту ж букву: " << func(text) << endl;
36
37     return 0;
38 }

```

Завдання 5

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  const int MAX_BOOKS = 5;
7  string books[MAX_BOOKS] = {"Book1", "Book2", "Book3", "Book4", "Book5"};
8  bool available[MAX_BOOKS] = {true, true, true, true, true};
9
10 void showBooks() {
11     cout << "Список книг:" << endl;
12     for (int i = 0; i < MAX_BOOKS; i++) {
13         cout << i + 1 << ". " << books[i] << " - " << (available[i] ? "доступна" : "уже взята") << endl;
14     }
15 }
16
17 void updateBookStatus(bool borrowing) {
18     int bookNum;
19     cout << "Введіть номер книги, яку хочете " << (borrowing ? "взяти: " : "повернути: ");
20     cin >> bookNum;
21
22     if (bookNum < 1 || bookNum > MAX_BOOKS) {
23         cout << "Неправильний номер книги. Спробуйте ще раз" << endl;
24         return;
25     }
26
27     if (available[bookNum - 1] == borrowing) {
28         available[bookNum - 1] = !borrowing;
29         cout << "Ви успішно " << (borrowing ? "взяли" : "повернули") << " книгу: " << books[bookNum - 1] << endl;
30     } else {
31         cout << "Ця книга вже " << (borrowing ? "зайнята" : "доступна") << endl;
32     }
33 }
34
35 int main() {
36     int choice;
37     do {
38         cout << endl << "Меню:" << endl;
39         cout << "1. Показати всі книги" << endl;
40         cout << "2. Взяти книгу" << endl;
41         cout << "3. Повернути книгу" << endl;
42         cout << "4. Вийти" << endl;
43         cout << "Виберіть опцію: ";
44         cin >> choice;
45
46         switch (choice) {
47             case 1: showBooks(); break;
48             case 2: updateBookStatus(true); break;
49             case 3: updateBookStatus(false); break;
50             case 4: cout << "Вихід з програми" << endl; break;
51             default: cout << "Неправильний вибір. Спробуйте ще раз" << endl;
52         }
53     } while (choice != 4);
54
55     return 0;
56 }

```

Завдання 6

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int n;
7      cin >> n;
8      string upper, lower;
9      cin >> upper >> lower;
10
11     for (int i = 0; i < n; ++i) {
12         if (upper[i] == '1' && lower[i] == '1') {
13             cout << "Yes" << endl;
14             return 0;
15         }
16     }
17
18     cout << "No";
19     return 0;
20 }

```

4) Результат виконання завдань, тестування та фактично затрачений час:

Завдання 1:

```

Сума перших 13 членів ряду: 2.01092
[Done] exited with code=0 in 1.087 seconds

```

Фактично затрачений час: 25 хв.

Завдання 2:

X = 0.1	SN = 0.0996687	SE = 0.0996667	Y = 0.0996687
X = 0.19	SN = 0.187762	SE = 0.187714	Y = 0.187762
X = 0.28	SN = 0.273009	SE = 0.273027	Y = 0.273009
X = 0.37	SN = 0.35438	SE = 0.354367	Y = 0.35438
X = 0.46	SN = 0.431139	SE = 0.431154	Y = 0.431139
X = 0.55	SN = 0.502843	SE = 0.502818	Y = 0.502843
X = 0.64	SN = 0.569313	SE = 0.569374	Y = 0.569313
X = 0.73	SN = 0.630578	SE = 0.630535	Y = 0.630578
X = 0.82	SN = 0.686818	SE = 0.68686	Y = 0.686818
X = 0.91	SN = 0.738309	SE = 0.738267	Y = 0.738313
X = 1	SN = 0.779149	SE = 0.785348	Y = 0.785398

Фактично затрачений час: 35 хв.

Завдання 3:

```

3 4 5
Кути трикутника: A = 36.8699, B = 53.1301, C = 90
3 3 3 3 3 3 3 3 3
Кути правильного 9-кутника: 140 140 140 140 140 140 140 140 140
10 10 10 10 10 10 10 10 10 10 10
Кути правильного 11-кутника: 147.273 147.273 147.273 147.273 147.

```

Фактично затрачений час: 60 хв.

Завдання 4:

```

Кількість від'ємних елементів: 3
Кількість слів, що починаються та закінчуються на ту ж букву: 2

```

Фактично затрачений час: 40 хв.

Завдання 5:

Меню:

1. Показати всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти

Виберіть опцію: 1

Список книг:

1. Book1 – доступна
2. Book2 – доступна
3. Book3 – доступна
4. Book4 – доступна
5. Book5 – доступна

Меню:

1. Показати всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти

Виберіть опцію: 2

Введіть номер книги, яку хочете взяти: 2

Ви успішно взяли книгу: Book2

Меню:

1. Показати всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти

Виберіть опцію: 1

Список книг:

1. Book1 – доступна
2. Book2 – уже взята
3. Book3 – доступна
4. Book4 – доступна
5. Book5 – доступна

Меню:

1. Показати всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти

Виберіть опцію: 3

Введіть номер книги, яку хочете повернути: 2

Ви успішно повернули книгу: Book2

Меню:

1. Показати всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти

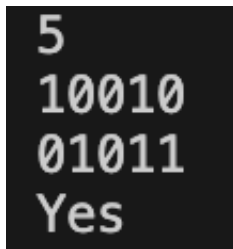
Виберіть опцію: 1

Список книг:

1. Book1 – доступна
2. Book2 – доступна
3. Book3 – доступна
4. Book4 – доступна
5. Book5 – доступна

Фактично затрачений час: 60 хв.

Завдання 6:



C++ 23	Зараховано	0.003	1.453
--------	-------------------	-------	-------

<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/71060#mySolutions>

Фактично затрачений час: 15 хв.

Робота з командою:



Висновок:

Під час виконання 3-го епіку я вивчив роботу з циклами в C++, зокрема вкладеними циклами та способами завершення їх виконання. Я засвоїв принципи створення та використання функцій, вивчив простір імен, перевантаження функцій та застосування функцій зі змінною кількістю параметрів. Окрім цього, я ознайомився з рекурсією та вбудованими функціями, що дозволило створювати більш ефективні та універсальні програми.