

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-13

Недосіка Назарій Вадимович

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

Мета: Розширити знання про цикли та їх види, навчитися використовувати функції крім `int main()`, використати це у написанні програм.

Теоретичні відомості:

- Лекції, практичні
- [w3schools.com](https://www.w3schools.com)
- ChatGPT
- Відео з YouTube

Виконання роботи

Завдання 1: Practice task

Задача

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.

Мета Задачі

Навчитися користуватися операторами циклів та функцією переходу на мітку:

1. `for() { ... }`
2. `for each`
3. `while() { ... }`
4. `do { ... } while()`
5. `go to`

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.

Розв'язок задачі:

```
1  #include <iostream>
2  #include <vector>
3  #include <string>
4
5  using namespace std;
6
7  vector<string> books = {"Кобзар", "Одіссея", "Іліада", "Перевтілення", "По той бік добра й зла"};
8  vector<bool> available = {true, true, true, true, true};
9
10 void listBooks() {
11     cout << "\nПерелік книг:\n";
12     for (int i = 0; i < books.size(); i++) {
13         cout << i + 1 << ". " << books[i] << (available[i] ? " (доступна)\n" : " (взята)\n");
14     }
15     cout << endl;
16 }
17
18 void borrowBook() {
19     int choice;
20
21     for (int i = 0; i < books.size(); i++) {
22         cout << i + 1 << ". " << books[i] << (available[i] ? " (доступна)\n" : " (взята)\n");
23     }
24
25     cout << "\nВведіть номер книги, яку бажаєте взяти: ";
26     cin >> choice;
27
28     if (choice < 1 || choice > books.size()) {
29         cout << "Неправильний вибір. Спробуйте ще раз.\n";
30         goto end;
31     }
32
33     if (available[choice - 1]) {
34         available[choice - 1] = false;
35         cout << "Ви успішно взяли книгу: " << books[choice - 1] << endl;
36     } else {
37         cout << "Цю книгу вже взяли.\n";
38     }
39
40 end;
41 }
42
```

```

43 void returnBook() {
44     int choice;
45
46     for (int i = 0; i < books.size(); i++) {
47         cout << i + 1 << ". " << books[i] << (available[i] ? " (доступна)\n" : " (взята)\n");
48     }
49
50     cout << "\nВведіть номер книги, яку бажаєте повернути: ";
51     cin >> choice;
52
53     if (choice < 1 || choice > books.size()) {
54         cout << "Неправильний вибір. Спробуйте ще раз.\n";
55         goto end;
56     }
57
58     if (!available[choice - 1]) {
59         available[choice - 1] = true;
60         cout << "Ви успішно повернули книгу: " << books[choice - 1] << endl;
61     } else {
62         cout << "Ця книга вже доступна.\n";
63     }
64
65     end;
66 }
67
68 int main() {
69     int choice;
70
71     do {
72         cout << "\nМеню бібліотеки:\n";
73         cout << "1. Переглянути всі книги\n";
74         cout << "2. Взяти книгу\n";
75         cout << "3. Повернути книгу\n";
76         cout << "4. Вийти\n";
77         cout << "Виберіть опцію: ";
78         cin >> choice;
79
80         switch (choice) {
81             case 1:
82                 listBooks();
83                 break;
84             case 2:
85                 borrowBook();
86                 break;
87             case 3:
88                 returnBook();
89                 break;
90             case 4:
91                 return 0;
92             default:
93                 cout << "Неправильний вибір \n";
94                 goto menu;
95         }
96
97     menu:
98         cout << "1. Головне меню\n";
99         cout << "2 .Вихід з програми\n";
100         cin >> choice;
101
102     } while (choice == 1);
103
104
105     return 0;
106 }

```

Вивід:

```

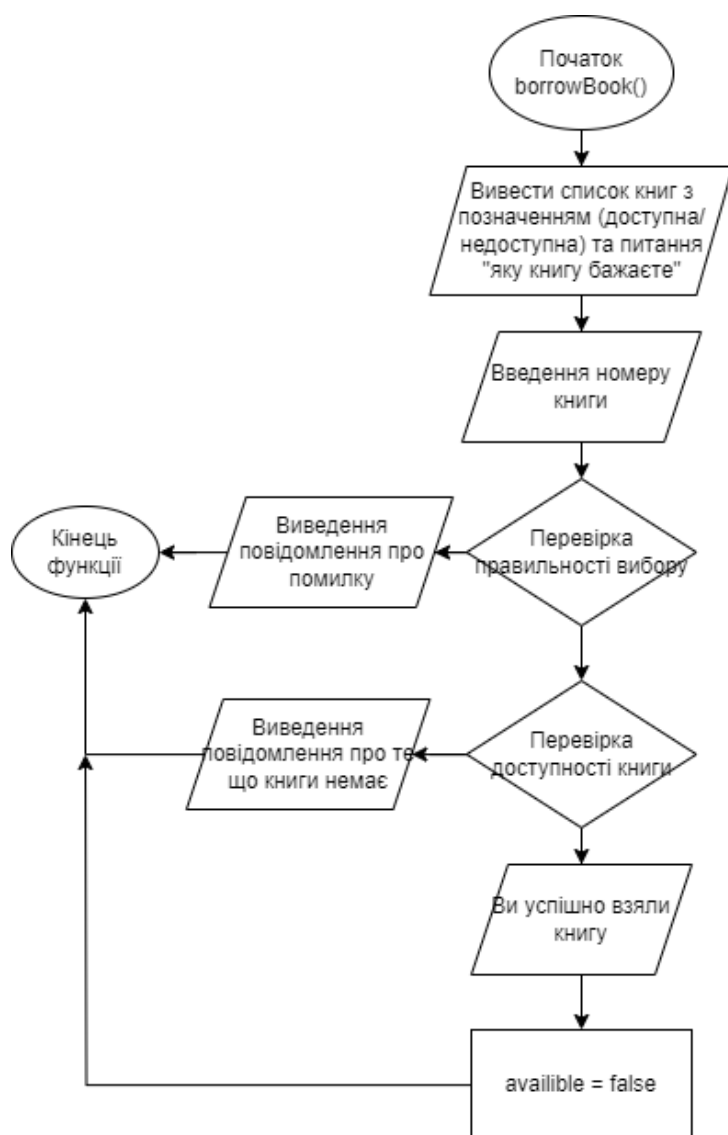
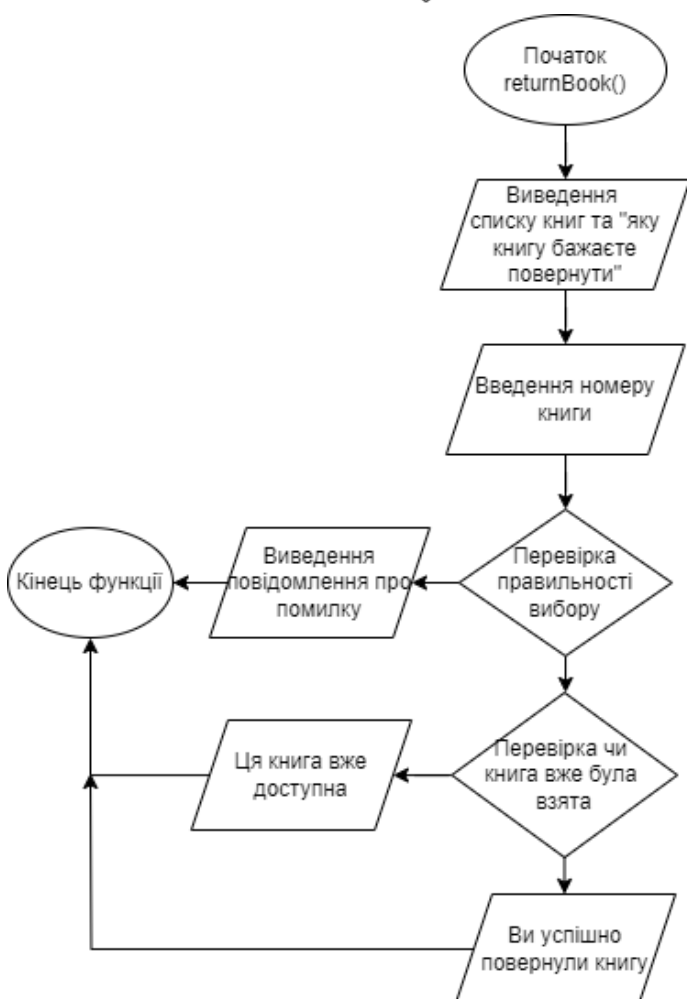
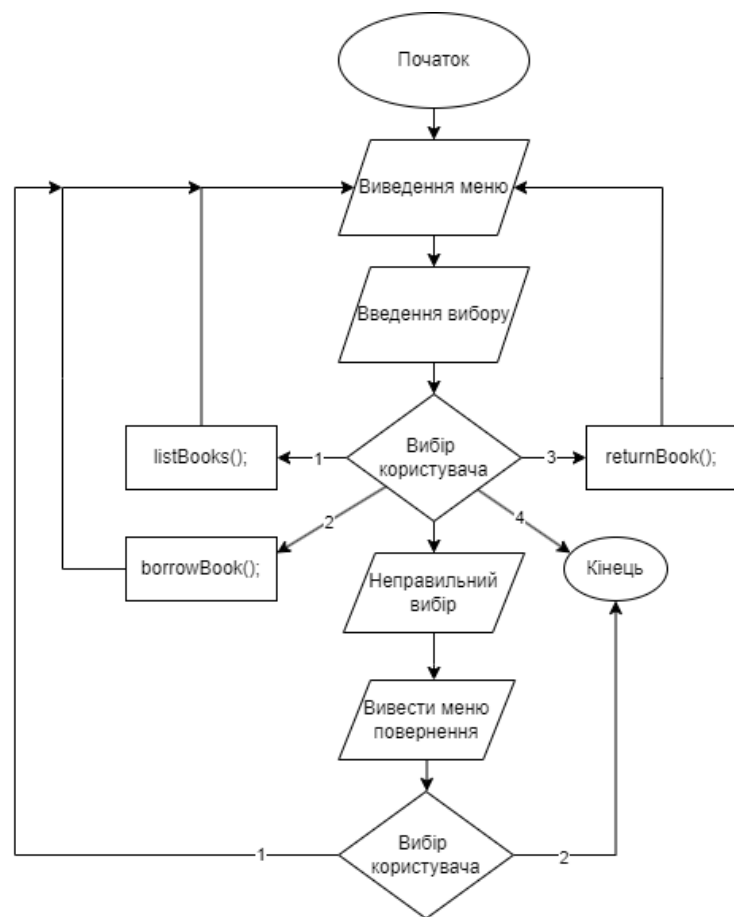
Меню бібліотеки:
1. Переглянути всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти
Виберіть опцію: 1

Список книг:
1. Кобзар (доступна)
2. Одиссея (доступна)
3. Іліада (доступна)
4. Перевтілення (доступна)
5. По той бік добра і зла (доступна)

1. Головне меню
2 .Вихід з програми

```

Блок схеми до задачі:



Час виконання завдання: ~ 4 години

Завдання 2: VNS Lab 2

Варіант 14

Знайти суму ряду з точністю $\varepsilon=0.0001$

Розв'язок задачі:

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      double epsilon = 0.0001;
8      double sum = 0.0;
9      int n = 1;
10
11     while (true) {
12         double factorial = 1.0;
13         for (int i = 1; i <= n; ++i) {
14             factorial *= i;
15         }
16
17         double an = factorial / (3 * pow(n, n));
18
19         sum += an;
20
21         if (abs(an) < epsilon) {
22             break;
23         }
24     }
25
26     n++;
27 }
28
29 cout << "Сума ряду з точністю  $\varepsilon=0.0001$  = " << sum;
30
31 return 0;
32 }
```

Вивід:

```
Сума ряду з точністю  $\varepsilon=0.0001$  = 0.626589
```

Час виконання завдання: ~ 30хв

Блок-схема до задачі:



Завдання 3: VNS Lab 3

Варіант 14

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 ($\varepsilon=0.0001$). Для порівняння знайти точне значення функції

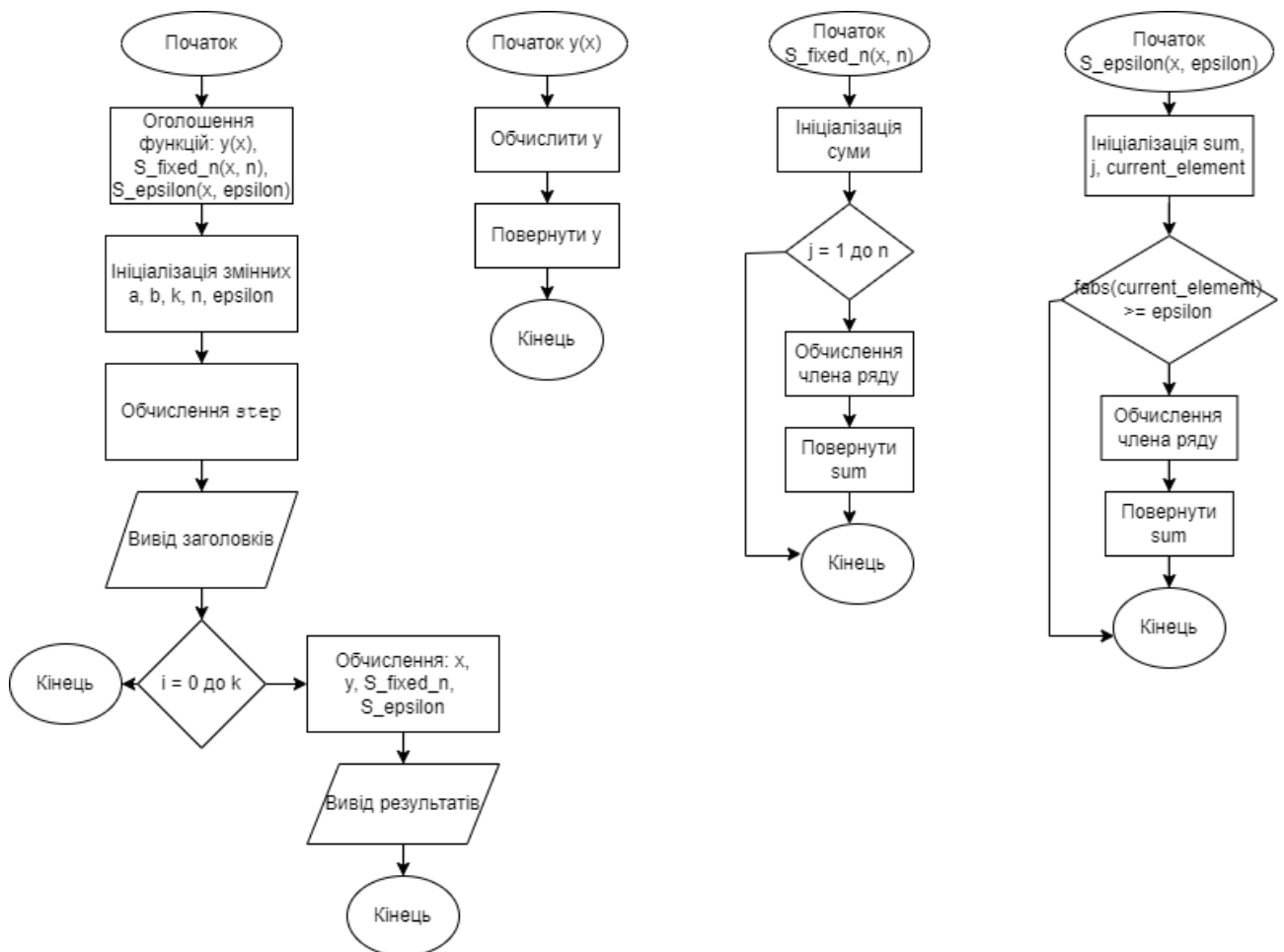
Розв'язок задачі:

```
1  #include <iostream>
2  #include <cmath>
3  #include <iomanip>
4
5  using namespace std;
6
7  double y(double x) {
8      return 0.25 * (x * x - M_PI * M_PI / 3);
9  }
10
11 double S_fixed_n(double x, int n) {
12     double sum = 0;
13     for (int j = 1; j <= n; ++j) {
14         sum = sum + pow(-1, j) * cos(j * x) / (j * j);
15     }
16     return sum;
17 }
18
19 double S_epsilon(double x, double epsilon) {
20     double sum = 0;
21     int j = 1;
22     double current_element;
23
24     do {
25         current_element = pow(-1, j) * cos(j * x) / (j * j);
26         sum = sum + current_element;
27         j = j + 1;
28     } while (fabs(current_element) >= epsilon);
29
30     return sum;
31 }
32
33 int main() {
34     double a = M_PI / 5;
35     double b = M_PI;
36     int k = 10;
37     int n = 20;
38     double epsilon = 0.0001;
39
40     double step = (b - a) / k;
41
42     cout << "x \t\t y_exact \t S_fixed_n \t S_epsilon \n";
43
44     for (int i = 0; i <= k; i = i + 1) {
45         double x = a + i * step;
46
47         double y_val = y(x);
48         double S_fixed_n_val = S_fixed_n(x, n);
49         double S_epsilon_val = S_epsilon(x, epsilon);
50
51         cout << fixed << setprecision(6) << x << "\t " << y_val << "\t " << S_fixed_n_val << "\t " << S_epsilon_val << "\n";
52     }
53
54     return 0;
55 }
```

Вивід:

x	y_exact	S_fixed_n	S_epsilon
0.628319	-0.723771	-0.722590	-0.723772
0.879646	-0.629023	-0.628103	-0.628794
1.130973	-0.502692	-0.503169	-0.502354
1.382301	-0.344778	-0.346309	-0.344373
1.633628	-0.155282	-0.156192	-0.154457
1.884956	0.065797	0.066869	0.066053
2.136283	0.318459	0.320894	0.317784
2.387610	0.602704	0.603857	0.604439
2.638938	0.918531	0.915548	0.920958
2.890265	1.265941	1.257435	1.269705
3.141593	1.644934	1.596163	1.635082

Блок-схема до задачі:



Час виконання завдання: ~ 1.5 години

Завдання 4: VNS Lab 7_1

Варіант 14

Написати функцію зі змінною кількістю параметрів для перетворення чисел із двійкової системи числення в трійкову. Написати викликаючу функцію main, що звертається до цієї функції не менше трьох разів з кількістю параметрів 3, 6, 7.

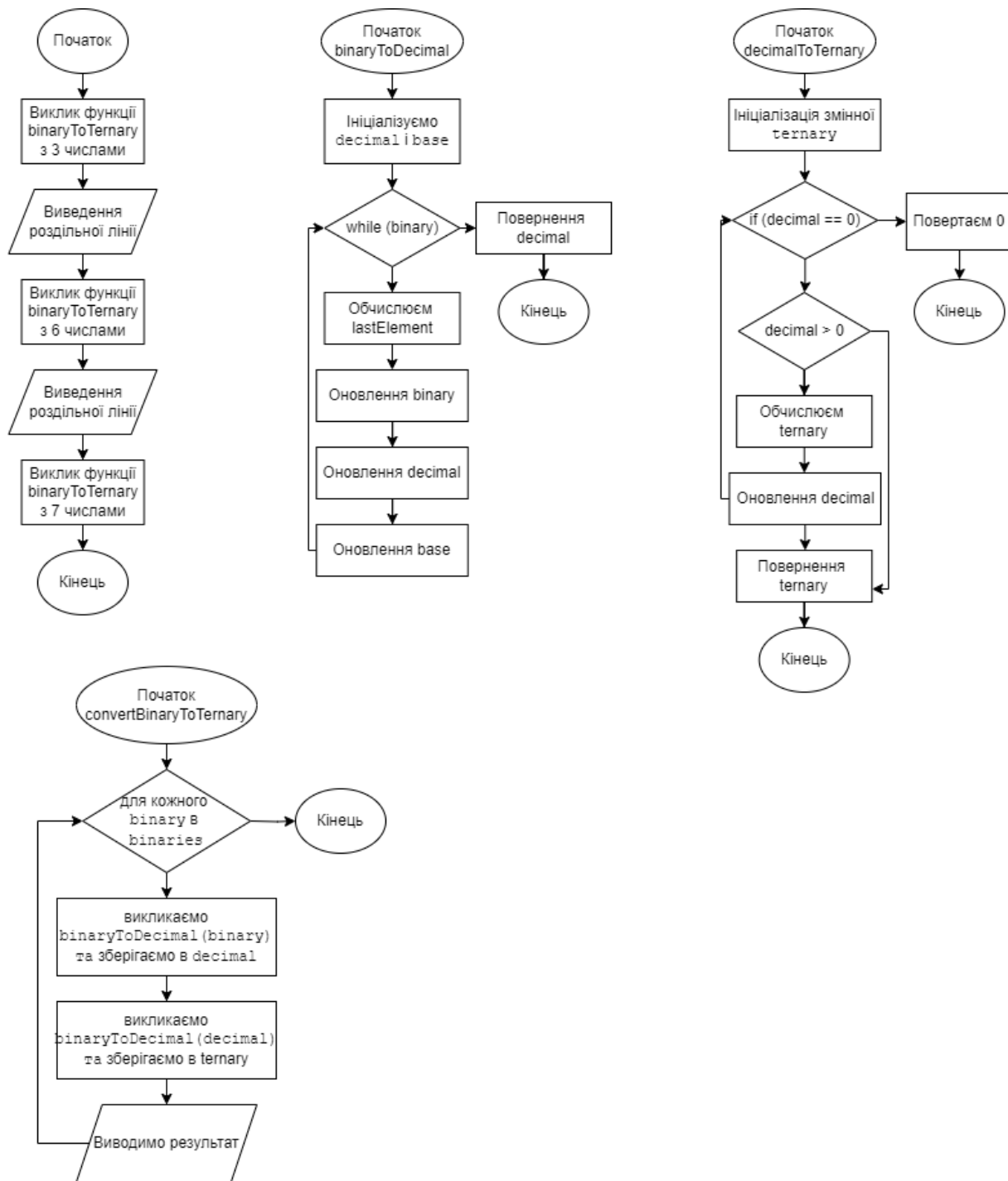
Розв'язок задачі:


```

1  #include <iostream>
2  #include <vector>
3  #include <string>
4
5  using namespace std;
6
7  int binaryToDecimal(int binary) {
8      int decimal = 0;
9      int base = 1;
10
11     while (binary) {
12         int lastElement = binary % 10;
13         binary /= 10;
14         decimal += lastElement * base;
15         base *= 2;
16     }
17
18     return decimal;
19 }
20
21 string decimalToTernary(int decimal) {
22     string ternary = "";
23
24     if (decimal == 0) {
25         return "0";
26     }
27
28     while (decimal > 0) {
29         ternary = to_string(decimal % 3) + ternary;
30         decimal /= 3;
31     }
32
33     return ternary;
34 }
35
36 void binaryToTernary(const vector<int>& binaries) {
37     for (int binary : binaries) {
38
39         int decimal = binaryToDecimal(binary);
40         string ternary = decimalToTernary(decimal);
41
42         cout << "Двійкове число: " << binary << " Трійкове число: " << ternary << endl;
43     }
44 }
45
46 int main() {
47     binaryToTernary({101, 110, 111});
48     cout << "-----" << endl;
49     binaryToTernary({1010, 1100, 1110, 1001, 1111, 1000});
50     cout << "-----" << endl;
51     binaryToTernary({10100, 11001, 11101, 10010, 10101, 11010, 11111});
52
53     return 0;
54 }

```

Блок-схема до задачі:



Вивід:

```

Двійкове число: 101 Трійкове число: 12
Двійкове число: 110 Трійкове число: 20
Двійкове число: 111 Трійкове число: 21
-----
Двійкове число: 1010 Трійкове число: 101
Двійкове число: 1100 Трійкове число: 110
Двійкове число: 1110 Трійкове число: 112
Двійкове число: 1001 Трійкове число: 100
Двійкове число: 1111 Трійкове число: 120
Двійкове число: 1000 Трійкове число: 22
-----
Двійкове число: 10100 Трійкове число: 202
Двійкове число: 11001 Трійкове число: 221
Двійкове число: 11101 Трійкове число: 1002
Двійкове число: 10010 Трійкове число: 200
Двійкове число: 10101 Трійкове число: 210
Двійкове число: 11010 Трійкове число: 222
Двійкове число: 11111 Трійкове число: 1011
  
```

Час виконання завдання: ~ 2.5 години

Завдання 4: VNS Lab 7_2

Варіант 14

Написати перевантажені функції й основну програму, що їх викликає.

а) для обчислення натурального логарифму;

б) для обчислення десяткового логарифму.

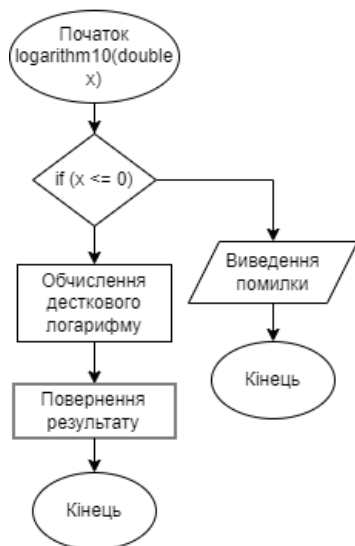
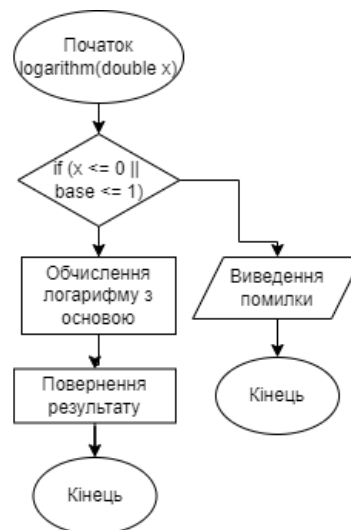
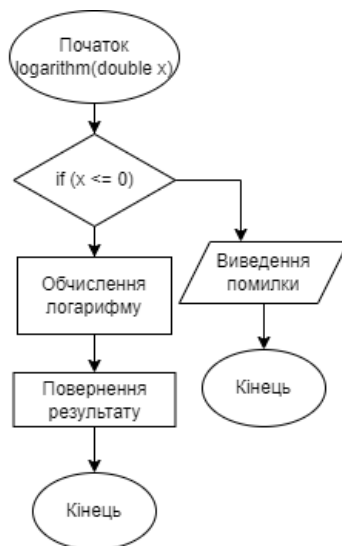
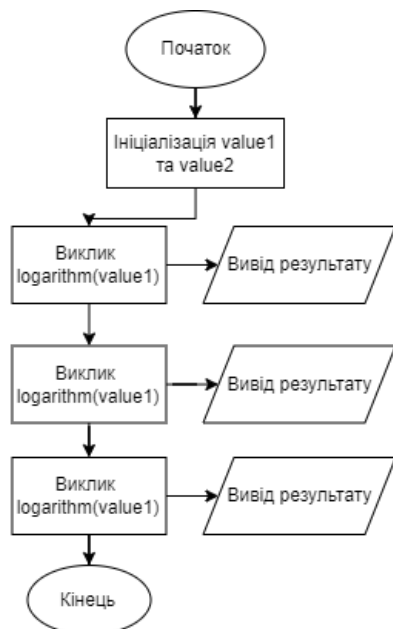
Розв'язок задачі:

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  double logarithm(double x) {
7      if (x <= 0) {
8          cout << "Помилка: натуральний логарифм визначений тільки для x > 0." << endl;
9          return 0;
10     }
11     return log(x);
12 }
13
14 double logarithm(double x, double base) {
15     if (x <= 0 || base <= 1) {
16         cout << "Помилка: натуральний логарифм визначений тільки для x > 0 i base > 1." << endl;
17         return 0;
18     }
19     return log(x) / log(base);
20 }
21
22 double logarithm10(double x) {
23     if (x <= 0) {
24         cout << "Помилка: десятковий логарифм визначений тільки для x > 0." << endl;
25         return 0;
26     }
27     return log10(x);
28 }
29
30 int main() {
31     double value1 = 10.0;
32     double value2 = 100.0;
33
34     cout << "Натуральний логарифм " << value1 << " = " << logarithm(value1) << endl;
35     cout << "Десятковий логарифм " << value2 << " = " << logarithm10(value2) << endl;
36     cout << "Логарифм " << value1 << " з основою 2 = " << logarithm(value1, 2) << endl;
37
38     return 0;
39 }
40
```

```
Натуральний логарифм 10 = 2.30259
Десятковий логарифм 100 = 2
Логарифм 10 з основою 2 = 3.32193
```

Вивід:

Блок-схема до задачі:



Час виконання завдання: ~ 1 година
Взаємодія з командою

Взаємодія з командою

<div> <div>IN PROGRESS</div> <div>7</div> <div>...</div> <div>+</div> <div>Add Task</div> </div>							
Name	Assignee	Due date	Priority	Status	Comments		
Danylo Kolbasiuk - Epic 4 $\cong 5/7$				IN PROGRE...		...	
Tofan Max - Epic 1 $\cong 0/14$			High	IN PROGRE...		...	
Nazar Nedosika - Epic 2 $\cong 4/5$			High	IN PROGRE...		...	
Tofan Max - Epic 2 $\cong 0/5$			High	IN PROGRE...		...	
Tofan Max - Epic 3 $\cong 0/7$			High	IN PROGRE...		...	
Nazar Nedosika - Epic 3 $\cong 4/7$			High	IN PROGRE...		...	
Alina Khodatska - Epic 3 $\cong 0/7$			High	IN PROGRE...		...	

Checklists 4/7

Checklist (4/7)

- ☒ VNS-Lab-2
- ☒ VNS-Lab-3
- ☒ VNS-Lab-7
- ☒ Class-Practice-Task
- ☐ Self-Practice Task
- ☐ Report
- ☐ Evaluation

+ New checklist item



Висновок: у цьому блоці я глибше працював з різними видами циклів та з функціями (еліпсис, рекурсія). Я використав це для виконання завдань цього епіку.