

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 ШІ-12
Стик Назарій Олегович

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:

Ознайомлення з основами використання циклічних конструкцій, включаючи вкладені цикли та способи завершення їх виконання. Вивчення функцій, простору імен, а також перевантаження функцій. Особливу увагу приділено функціям із змінною кількістю параметрів (еліпсис) та рекурсії. Додатково досліджуються можливості вбудованих функцій і їх роль в оптимізації коду.

Теоретичні відомості:

У даній роботі розглядаються цикли та вкладені цикли, які використовуються для повторюваних дій. Також вивчаються способи завершення виконання циклів за допомогою break та continue. Досліджується використання функцій, їх перевантаження, функції з змінною кількістю параметрів (еліпсис), а також рекурсія. Крім цього, аналізуються вбудовані функції та поняття простору імен для організації коду.

Джерела:

- Декілька відео на YouTube:
 - Цикл while - <https://www.youtube.com/watch?v=zPi163Xo4m8&t=12s>
 - Цикл do while - <https://www.youtube.com/watch?v=AFTfqXdrYBQ>
 - Цикл for – 1) <https://www.youtube.com/watch?v=CJm6o82Vu2g>
2) https://www.youtube.com/watch?v=v_ZdslCeRfQ
- Певну інформацію брав на сайтах:
 - <http://cpp.dp.ua/operator-y-tsyklu/>
 - <https://acode.com.ua/urok-72-tsykl-for/>
- Також вивчав як працюють цикли за допомогою ChatGPT.

Виконання роботи:

- **Завдання №1:** Theory Education Activities

Очікувано часу: 3 дні.

Витрачено часу: 2 дні.

- **Завдання №2:** Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)

Очікувано часу: 2 години.

Витрачено часу: 4 години.

- **Завдання №3:** Lab# programming: VNS Lab 2

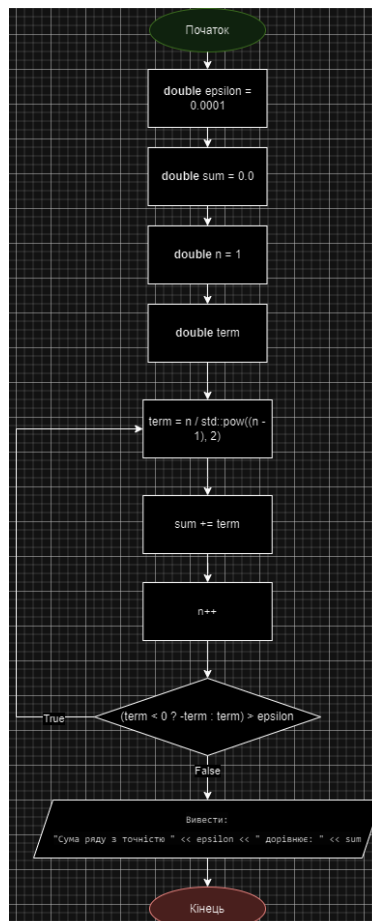
Код:

```

1  #include <iostream>
2  #include <cmath> // для pow()
3
4  int main() {
5      double epsilon = 0.0001;
6      double sum = 0.0;
7      double n = 1;
8      double term;
9
10     do {
11         term = n / std::pow((n - 1), 2);
12
13         sum += term;
14         n++;
15     } while ((term < 0 ? -term : term) > epsilon);
16
17     std::cout << "Сума ряду з точністю " << epsilon << " дорівнює: " << sum << std::endl;
18
19     return 0;
20 }

```

Блок-схема:



Очікувано часу: **30 хвилин.**

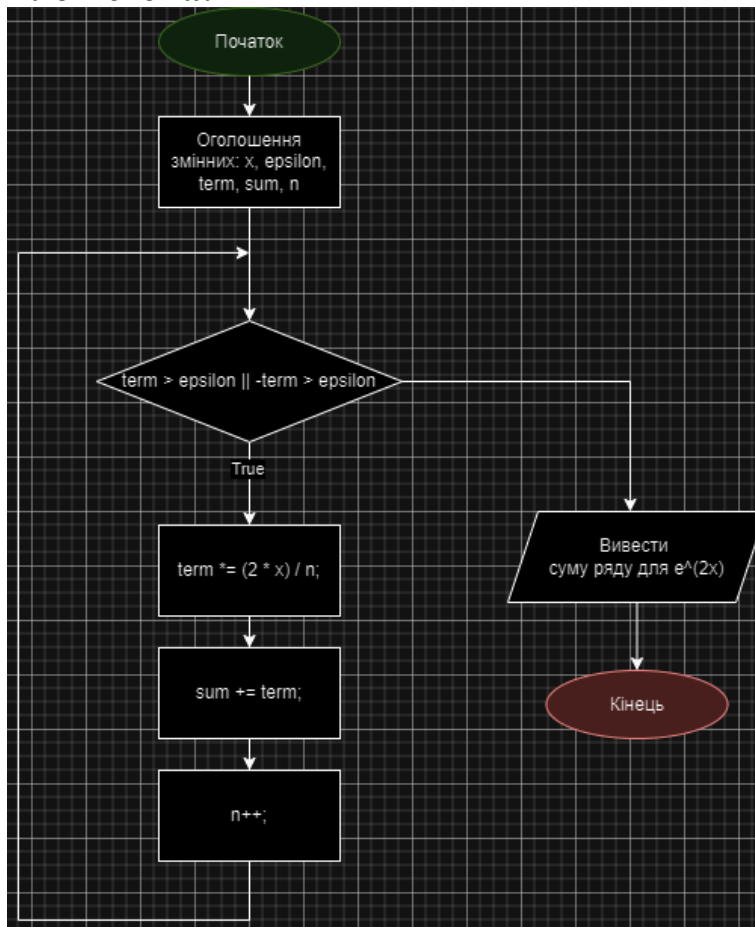
Витрачено часу: **30 хвилин.**

- Завдання №4: Lab# programming: VNS Lab 3

Код:

```
1 // Варіант 19
2 #include <iostream>
3
4 int main()
5 {
6     double x = 1.0;
7     double epsilon = 0.000001;
8     double term = 1.0;
9     double sum = term;
10    int n = 1;
11
12    // Обчислюємо суму ряду доти, доки доданок більше за точність
13    while (term > epsilon || -term > epsilon)
14    {
15        term *= (2 * x) / n;
16        sum += term;
17        n++; // збільшуємо факторіал
18    }
19
20    std::cout << "Сума ряду для  $e^{(2x)}$  при  $x =$ " << x << " дорівнює: " << sum << std::endl;
21
22    return 0;
23 }
```

Блок-схема:



Очікувано часу: **30 хвилин.**

Витрачено часу: **30 хвилин.**

Завдання №5: Lab# programming: VNS Lab 7

Task 1:

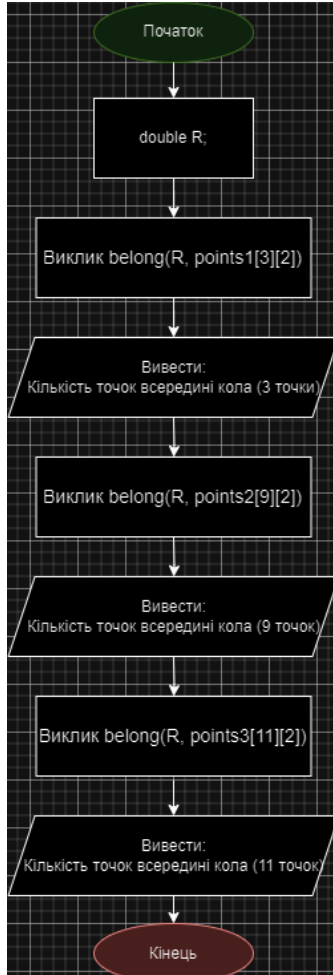
Код:

```

1  #include <iostream>
2  #include <cmath>
3  #include <vector>
4
5  // Функція, що перевіряє, чи належить точка (x, y) колу з радіусом R
6  bool isInsideCircle(double x, double y, double R)
7  {
8      return (x * x + y * y) <= (R * R);
9  }
10
11 // Функція, що визначає кількість точок, які належать колу
12 int belong(double R, double points[][2], int numPoints)
13 {
14     int count = 0;
15     for (int i = 0; i < numPoints; ++i)
16     {
17         double x = points[i][0];
18         double y = points[i][1];
19         if (isInsideCircle(x, y, R))
20         {
21             ++count;
22         }
23     }
24     return count;
25 }
26
27 int main()
28 {
29     double R = 5.0;
30
31     // Масив з 3 точками
32     double points1[3][2] = {{1.0, 2.0}, {3.0, 4.0}, {5.0, 6.0}};
33     std::cout << "Точок всередині кола (3 точки): " << belong(R, points1, 3) << std::endl;
34
35     // Масив з 9 точками
36     double points2[9][2] = {{1.0, 2.0}, {3.0, 4.0}, {5.0, 6.0}, {7.0, 8.0}, {9.0, 10.0}, {11.0, 12.0}, {13.0, 14.0}, {15.0, 16.0}, {17.0, 18.0}};
37     std::cout << "Точок всередині кола (9 точок): " << belong(R, points2, 9) << std::endl;
38
39     // Масив з 11 точками
40     double points3[11][2] = {{1.0, 2.0}, {3.0, 4.0}, {5.0, 6.0}, {7.0, 8.0}, {9.0, 10.0}, {11.0, 12.0}, {13.0, 14.0}, {15.0, 16.0}, {17.0, 18.0}, {19.0, 20.0}, {21.0, 22.0}};
41     std::cout << "Точок всередині кола (11 точок): " << belong(R, points3, 11) << std::endl;
42
43     return 0;
44 }

```

Блок-схема:

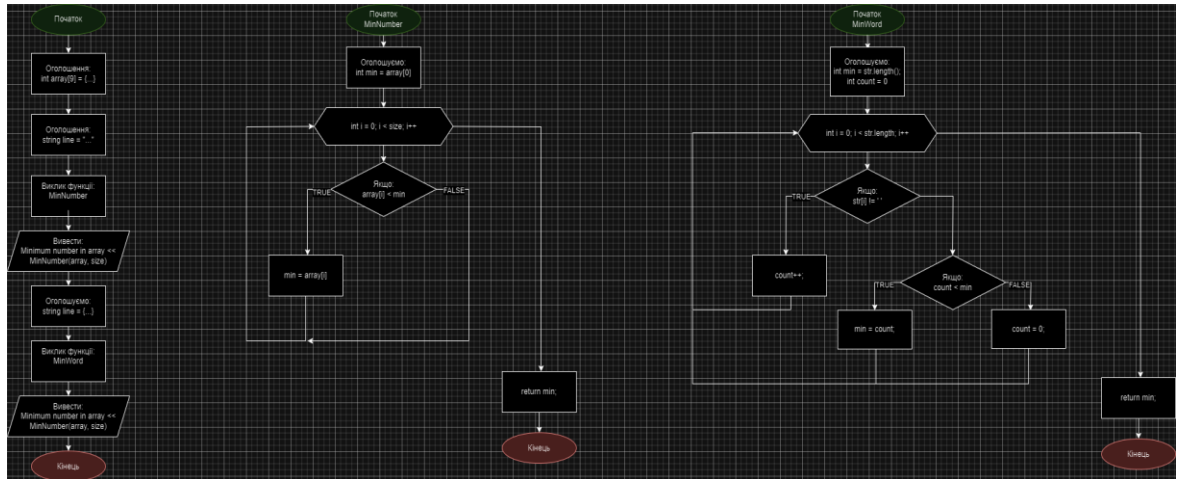


Task 2:

Код:

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  // Розв'язати зазначене у варіанті завдання, використовуючи функції зі змінною кількістю параметрів.
7  // а) для масиву цілих чисел знаходить мінімальний елемент;
8  // б) для рядка знаходить довжину найкоротшого слова .
9
10 int MinNumber(int array[], int size)
11 {
12     int min = array[0];
13
14     for (int i = 0; i < size; i++)
15     {
16         if (array[i] < min)
17         {
18             min = array[i];
19         }
20     }
21     return min;
22 }
23
24 int MinWord(string str)
25 {
26     int min = str.length();
27     int count = 0;
28
29     for (int i = 0; i < str.length(); i++)
30     {
31         if (str[i] != ' ')
32         {
33             count++;
34         }
35         else
36         {
37             if (count < min)
38             {
39                 min = count;
40             }
41             count = 0;
42         }
43     }
44     return min;
45 }
46
47 int main()
48 {
49     int array[9] = {3, 15, 7, 12, 19, 5, 8, 2, 11};
50     int size = sizeof(array) / sizeof(array[0]);
51     cout << "Minimum number in array: " << MinNumber(array, size) << endl;
52
53     string line = "This is a test string";
54     cout << "Length of the shortest word: " << MinWord(line) << endl;
55
56     return 0;
57 }
```

Блок-схема:



Очікувано часу: 1 година.

Витрачено часу: 2 години.

Завдання №6: Practice# programming: Class Practice Task

Код:

```

1  #include <iostream>
2  #include <vector>
3  #include <string>
4
5  using namespace std;
6
7  void ListBooks(const vector<string>& books, const vector<bool>& availability)
8  {
9      cout << "Books in the library:\n";
10     for (int i = 0; i < books.size(); ++i)
11     {
12         cout << i + 1 << ". " << books[i] << " - " << (availability[i] ? "Available" : "Not Available") << endl;
13     }
14 }
15
16 void BorrowBook(vector<bool>& availability)
17 {
18     int bookNumber;
19     cout << "Enter the book number to borrow: ";
20     cin >> bookNumber;
21     if (bookNumber > 0 && bookNumber <= availability.size() && availability[bookNumber - 1])
22     {
23         availability[bookNumber - 1] = false;
24         cout << "You have borrowed the book.\n";
25     }
26     else
27     {
28         cout << "Invalid book number or the book is not available.\n";
29     }
30 }
31
32 void ReturnBook(vector<bool>& availability) {
33     int bookNumber;
34     cout << "Enter the book number to return: ";
35     cin >> bookNumber;
36     if (bookNumber > 0 && bookNumber <= availability.size() && !availability[bookNumber - 1])
37     {
38         availability[bookNumber - 1] = true;
39         cout << "You have returned the book.\n";
40     }
41     else
42     {
43         cout << "Invalid book number or the book was not borrowed.\n";
44     }
45 }
46
47 int main()
48 {
49     vector<string> books = {"Book1", "Book2", "Book3", "Book4", "Book5"};
50     vector<bool> availability(books.size(), true);
51     char choice;
52
53     do {
54         menu:
55         cout << "Library Management System\n";
56         cout << "1. List all books\n";

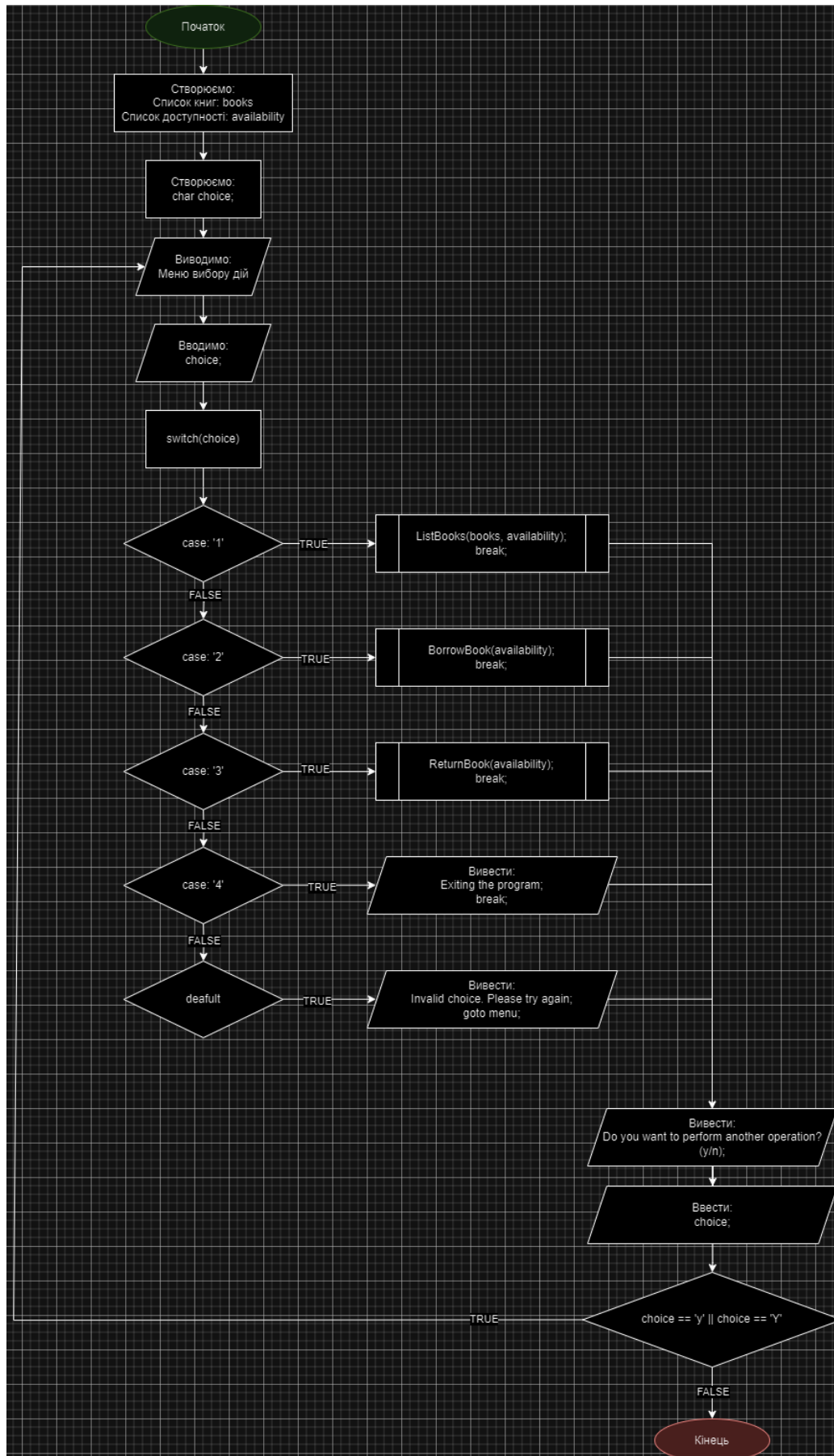
```

```

55     cout << "Library Management System\n";
56     cout << "1. List all books\n";
57     cout << "2. Borrow a book\n";
58     cout << "3. Return a book\n";
59     cout << "4. Exit\n";
60     cout << "Enter your choice: ";
61     cin >> choice;
62
63     switch (choice)
64     {
65         case '1':
66             ListBooks(books, availability);
67             break;
68         case '2':
69             BorrowBook(availability);
70             break;
71         case '3':
72             ReturnBook(availability);
73             break;
74         case '4':
75             cout << "Exiting the program.\n";
76             break;
77         default:
78             cout << "Invalid choice. Please try again.\n";
79             goto menu;
80     }
81
82     cout << "Do you want to perform another operation? (y/n): ";
83     cin >> choice;
84     } while (choice == 'y' || choice == 'Y');
85
86     return 0;
87 }

```


Блок-схема:



Очікувано часу: **2 години.**

Витрачено часу: **3 години.**

- **Завдання №7:** Practice# programming: Self Practice Task

Код:

```
1  #include <iostream>
2
3  int main()
4  {
5      long long a, b, c;
6
7      // Апельсини Марічки, Софійки і Петрика
8      std::cin >> a >> b >> c;
9
10     // Суми апельсинів Марічки та Софійки
11     long long sum = a + b;
12
13     if (sum > c)
14     {
15         std::cout << "YES" << std::endl;
16     }
17     else
18     {
19         std::cout << "NO" << std::endl;
20     }
21
22     return 0;
23 }
```

Задача на Алготестері:

<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40231#mySolutions>

Очікувано часу: **20 хвилин.**

Витрачено часу: **10 хвилин.**

Pull-Request: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/114

Висновок: Під час роботи я ознайомився з основами циклів, таких як for(), while(), do { } while(), та оператором переходу goto. Я розібрався, як працюють ці конструкції, включаючи вкладені цикли та способи їх завершення. Також вивчив функції, простір імен, та перевантаження функцій для різних завдань. Окремо зосередився на функціях зі змінною кількістю параметрів (еліпсис) та рекурсії. Дослідив, як вбудовані функції можуть покращити ефективність коду.