

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:**

Практичних Робіт до блоку № 3

**Виконав:**

Студент групи ІІІ-11

Голейчук Іван Миколайович

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:** Ознайомитися з основами циклів, дослідити концепцію функцій у C++, навчитися зображати функції в блок схемах, розібратися зі способами завершення виконання циклів.

**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++: Індивідуальний план опрацювання теорії

## **Індивідуальний план опрацювання теорії:**

**Тема №1: Введення в Цикли та їх Види в C++**

**Джерела інформації:**

- Лекції Олександра Пшеничного;
- Практичні заняття;
- Використання штучного інтелекту (чат gpt);
- Youtube.

**Що опрацьовано:** Ознайомився з основними видами циклів у C++: for, while та do-while. Розібрався з їхніми відмінностями та прикладами використання для ітераційних завдань.

**Тема №2: Управління Виконанням Циклів**

**Джерела інформації:**

- Лекції Олександра Пшеничного;
- Практичні заняття;

-Використання штучного інтелекту (чат gpt);

-Youtube.

**Що опрацьовано:** Вивчив способи управління виконанням циклів, зокрема оператори break, continue, і return, а також їх застосування для контролю і виходу з циклів за певних умов.

### **Тема №3: Вкладені Цикли**

#### **Джерела інформації:**

-Лекції Олександра Пшеничного;

-Практичні заняття;

-Використання штучного інтелекту (чат gpt);

-Youtube.

**Що опрацьовано:** Розглянув вкладені цикли (цикли всередині інших циклів) і навчився їх використовувати, особливо для роботи з багатовимірними масивами чи повторюваних обчислень.

### **Тема №4: Основи Функцій у C++**

#### **Джерела інформації:**

-Лекції Олександра Пшеничного;

-Практичні заняття;

-Використання штучного інтелекту (чат gpt);

-Youtube.

**Що опрацьовано:** Вивчив основи функцій: оголошення, виклик, повернення значень і параметри функцій. Зрозумів, як функції структурують код і дозволяють повторно використовувати логіку.

### **Тема №5: Перевантаження Функцій та Простір Імен**

#### **Джерела інформації:**

-Лекції Олександра Пшеничного;

-Практичні заняття;

-Використання штучного інтелекту (чат gpt);

-Youtube.

**Що опрацьовано:** Розібрався з перевантаженням функцій, тобто можливістю створювати функції з однаковою назвою, але різними параметрами. Також опрацював простір імен (namespace), щоб організувати код і уникати конфліктів імен.

## **Тема №6: Розширені Можливості Функцій**

### **Джерела інформації:**

- Лекції Олександра Пшеничного;
- Практичні заняття;
- Використання штучного інтелекту (чат gpt);
- Youtube.

**Що опрацьовано:** Розглянув розширені можливості функцій у C++, такі як параметри за замовчуванням, передача за посиланням і значенням, а також використання функцій як аргументів інших функцій.

## **Тема №7: Вбудовані Функції в C++**

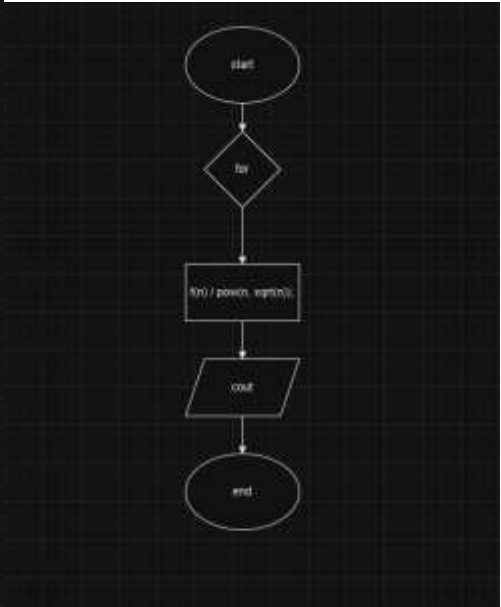
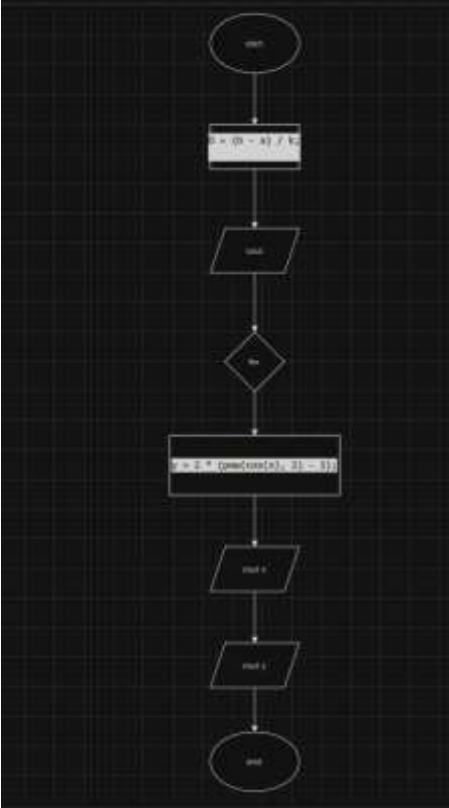
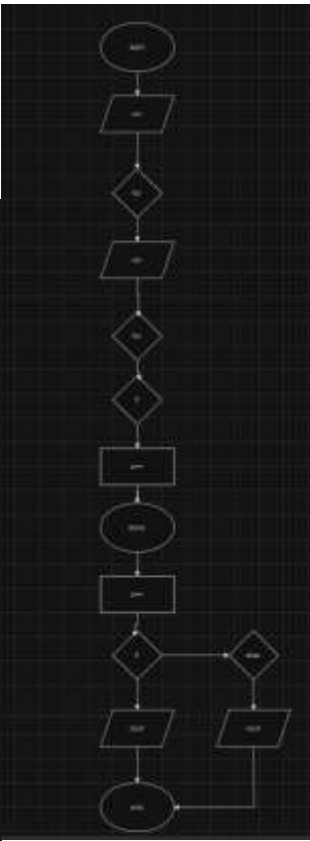
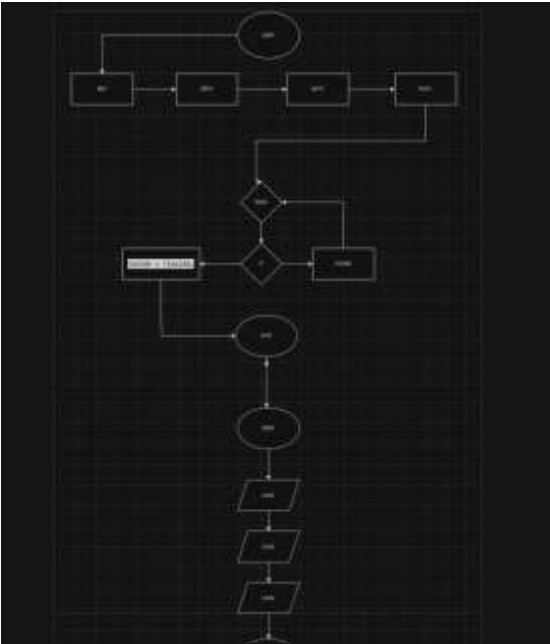
### **Джерела інформації:**

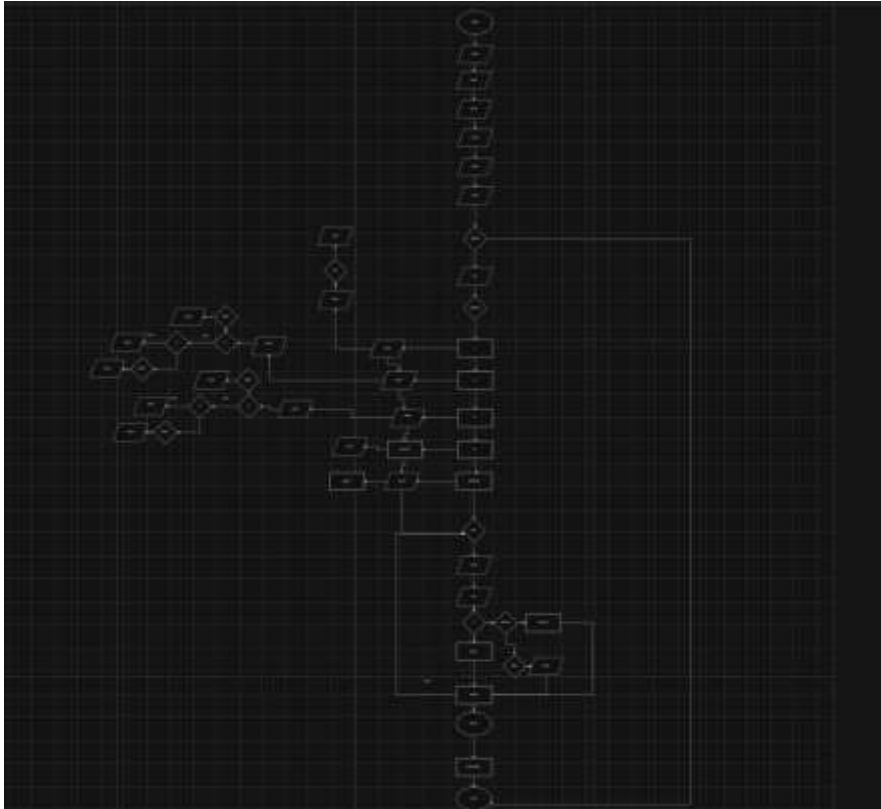
- Лекції Олександра Пшеничного;
- Практичні заняття;
- Використання штучного інтелекту (чат gpt);
- Youtube.

**Що опрацьовано:** Склав індивідуальний план для опрацювання теорії про вбудовані функції в C++, включаючи математичні функції, функції для роботи з рядками та функції введення/виведення, щоб зрозуміти їхнє практичне застосування.

## **Виконання роботи:**

**Task 2** - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)





Витрачено часу: 5 годин.

### Task 3 - Lab# programming: VNS Lab 2

23) Знайти суму 10 членів ряду, у якому

$$a_n = \frac{n!}{n^{\sqrt{n}}}$$

```

1: #include <iostream>
2: #include <cmath>
3:
4: using namespace std;
5:
6:
7: |
8: double f(int n) {
9:     double r = 1;
10:    for (int i = 1; i <= n; ++i) {
11:        r *= i;
12:    }
13:    return r;
14: }
15:
16: int main() {
17:
18:     double suma = 0.0;
19:
20:     for (int n = 1; n <= 10; ++n) {
21:         double suma = f(n) / pow(n, sqrt(n));
22:     }
23:
24:     cout << suma << endl;
25:
26:     return 0;
27: }
28:
29:

```

Витрачено часу: 1 година.

#### Task 4 - Lab# programming: VNS Lab 3

23	$y = 2(\cos^2 x - 1)$	$0,1 \leq x \leq 1$
----	-----------------------	---------------------



```

1  #include <iostream>
2  #include <cmath>
3  #include <iomanip>
4
5  using namespace std;
6
7  int main() {
8
9      double a = 0.1, b = 1.0;
10
11     int k = 10;
12
13     double D = (b - a) / k;
14
15
16     cout << "y = 2 * (cos^2(x) - 1);" << endl;
17
18
19     for (double x = a; x <= b; x += D) {
20
21         double y = 2 * (pow(cos(x), 2) - 1);
22
23         cout << "x = " << x << endl;
24         cout << "y = " << y << endl;
25     }
26
27     return 0;
28 }
29

```

Витрачено часу: 2 години

## Task 5 - Lab# programming: VNS Lab 7

23. Написати функцію (або макровизначення), що знаходить довжину сторони за координатами його точок.. Написати функцію `belong`, що визначає чи належить точка  $M$  з координатами  $(x, y)$  трикутнику, заданому координатами вершин. Написати функцію з змінною кількістю параметрів, що визначає чи належить точка  $M$  опуклому багатокутнику, заданому координатами своїх вершин.

```

1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 double f(double x1, double y1, double x2, double y2) {
7     return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));
8 }
9
10 bool belong(double x, double y, double x1, double y1, double x2, double y2, double x3, double y3) {
11
12     double abc = abs((x2 - y3) + x3 * (y3 - y1) + x1 * (y1 - y2)) / 2.0;
13     double abn = abs((x2 - y) + x3 * (y - y1) + x1 * (y1 - y2)) / 2.0;
14     double acn = abs((x2 - y) + x3 * (y3 - y1) + x1 * (y1 - y2)) / 2.0;
15     double bcn = abs((x2 * (y - y1) + x3 * (y3 - y2) + x * (y2 - y3)) / 2.0);
16
17     return (abn + acn == abc);
18 }
19
20 bool p(double x, double y, double vertices[12], int numVertices) {
21     bool inside = false;
22
23     for (int i = 0, j = numVertices - 1; i < numVertices; j = i++) {
24         if ((vertices[i][0] > y) && (vertices[j][0] > y) && x < ((vertices[j][0] - vertices[i][0]) * (y - vertices[i][0]) / (vertices[j][0] - vertices[i][0]) + vertices[i][0])) {
25             inside = true;
26         }
27     }
28     return inside;
29 }
30
31 int main() {
32     double ab = {1, 1, 4, 1};
33     cout << "ab: " << ab << endl;
34
35     bool t = belong(2, 2, 1, 1, 4, 1, 2, 4);
36
37     cout << "H(2, 2) ∈ ABC? " << (t ? "Yes" : "No") << endl;
38
39     double w[12] = {1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0};
40     bool p = p(2, 2, w, 4);
41
42     cout << "Does the point H(2, 2) belong to a polygon? " << (p ? "Yes" : "No") << endl;
43
44     return 0;
45 }

```

Витрачено часу: 4 години.

## Task 6 - Practice# programming: Class Practice Task

```

1 //include libraries
2 #include <iostream>
3 #include <vector>
4 #include <string>
5
6 using namespace std;
7
8 //display menu
9 void d() {
10
11     cout << "Use library menu now" << endl;
12
13     cout << "Show all books" << endl;
14
15     cout << "Take a book" << endl;
16
17     cout << "Return the book" << endl;
18
19     cout << "Exit" << endl;
20
21     cout << "Select an option: " << endl;
22 }
23
24 int main() {
25
26     vector<string> b = { "Book 1", "Book 2", "Book 3", "Book 4" }; //books
27     vector<bool> a = { true, true, true, true }; //available
28
29     int q; //choice
30     bool w = false; //exit
31
32     while (true) {
33         d();
34         cin >> q;
35
36         switch (q) {
37
38             case 1: {
39
40                 cout << "List of all books:" << endl;
41                 for (int i = 0; i < b.size(); i++) {
42                     cout << i + 1 << " : " << b[i] << " (available) : " << (a[i]) << endl;
43                 }
44                 break;
45             } case 2: {
46
47                 int b1; //book's number
48
49                 cout << "Enter the number of the book you want to take: " << endl;
50                 cin >> b1;
51
52                 if (b1 > 0 && b1 <= b.size()) {
53                     if (a[b1 - 1]) {
54                         a[b1 - 1] = false;
55                         cout << "You took the book: " << b[b1 - 1] << endl;
56                     }
57                     else {
58                         cout << "This book is already taken!" << endl;
59                     }
60                 }
61                 else {
62                     cout << "Invalid book number!" << endl;
63                 }
64                 break;
65             } case 3: {
66
67                 int r1; //number of book
68
69                 cout << "Select the book you want to return : " << endl;
70                 cin >> r1;
71
72                 if (r1 > 0 && r1 <= b.size()) {
73                     if (!a[r1 - 1]) {
74                         a[r1 - 1] = true;
75                         cout << "You returned the book: " << b[r1 - 1] << endl;
76                     }
77                     else {
78                         cout << "This book is now available!" << endl;
79                     }
80                 }
81                 else {
82                     cout << "Invalid book number!" << endl;
83                 }
84                 break;
85             } case 4: {
86
87                 w = true;
88
89                 cout << "Thank you for using the library!" << endl;
90                 break;
91             } default: {
92
93                 cout << "Wrong choice. Try again." << endl;
94                 goto start;
95             }
96         }
97
98         char c; //continue
99
100         do {
101
102             cout << "Do you want to perform another operation? (y/n): " << endl;
103             cin >> c;
104
105             if (c == 'y' || c == 'Y') {
106                 goto start;
107             }
108             else if (c == 'n' || c == 'N') {
109                 w = true;
110             }
111             else cout << "Wrong choice. ";
112
113         } while (c != 'y' && c != 'Y' && c != 'n' && c != 'N');
114
115         start:
116
117         continue;
118     }
119
120     return 0;
121 }

```

## Task 7 - Practice# programming: Self Practice Task

## Упорядкуй коней

Ліміт: 2 sec., 256 MB

У Зенка з Маричкою є  $n$  жеребів, пронумерованих від 1 до  $n$ , яких вони тримають у стайні. У стайні є  $n$  стійл, розташованих у ряд, також пронумерованих від 1 до  $n$ . Кінь з номером  $i$  має своє стійло з тим самим номером  $i$ . Одного разу Маричка після важкого робочого дня на фермі зводила коней до стайні, та випадково переплутала порядок коней. У стійло  $i$  вона поставила коня з номером  $p_i$ .

Зенка зауважило, що щось у стайні не так. Він хоче переставити коней на свої місця.

Вам задано число  $k$ .

Зенка спробує розділити коней в стайні на  $k$  проміжків. Кожен проміжок міститиме коней, що стоять у стійлах з номерами від  $l$  до  $r$ , тобто коней з номерами  $p_l, p_{l+1}, \dots, p_r$ , де  $1 \leq l \leq r \leq n$ . Кожен кінь ходитиме рівно до одного проміжку. Після цього Зенка зможе переставити проміжки між собою як завгодно.

Скажіть, чи зможе Зенка переставити коней у правильному порядку, тобто  $1, 2, \dots, n$ ?

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      ios::sync_with_stdio(false);
7      cin.tie(nullptr);
8
9      int n, k;
10     cin >> n >> k;
11     vector<int> p(n);
12
13     for (int i = 0; i < n; i++) {
14         cin >> p[i];
15     }
16
17     int s = 0;
18
19
20     for (int i = 0; i < n - 1; i++) {
21         if (p[i] + 1 != p[i + 1]) {
22             s++;
23         }
24     }
25     s++;
26
27
28     if (s <= k) {
29         cout << "Yes\n";
30     }
31     else {
32         cout << "No\n";
33     }
34
35     return 0;
36 }
37
```

Витрачено часу: 4 години.

**Робота у команді:**



Це останній епік в цій команді, і я дуже вдячний хлопцям за затрачений час, допомогу та роботу разом.

**Висновок:** Опрацювавши ці теми, я отримав ґрунтовне розуміння циклів і функцій у C++. Це дозволило мені освоїти різні типи циклів та способи управління їх виконанням, що є необхідним для побудови ефективних ітераційних структур. Знання вкладених циклів розширило мої можливості роботи з багатовимірними структурами, а функції допомогли краще структурувати код, зробивши його більш організованим і зручним для повторного використання.

Додатково я зрозумів, як працює перевантаження функцій і простір імен, що дозволяє уникати конфліктів між іменами та організовувати код. Завдяки вивченню розширених можливостей функцій я можу налаштовувати їхню поведінку за потреби. Індивідуальний план опрацювання вбудованих функцій допоможе мені використовувати їх у різних завданнях та підвищити ефективність свого коду.

Загалом, ці теми сформували основу для розуміння важливих концепцій у програмуванні, які будуть корисними для подальшого розвитку в алгоритмах та структурі даних.