

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  
Бісюк Роман Васильович

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

**Мета роботи:** Навчитися використовувати функції різних видів, цикли та виконувати завдання над ними. Також будувати блок-схеми складніших типів.

**Теоретичні відомості:**

- цикл for
- цикл while
- цикл do, while
- функції

Джерела:

- [aCode – перезавантаження функцій в C++](#)
- [aCode – еліпсис](#)
- [aCode – goto](#)
- [Harvard CS50 – теорія та задачі по циклах та рекурсії](#)

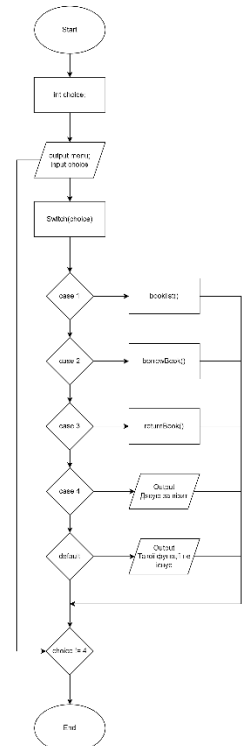
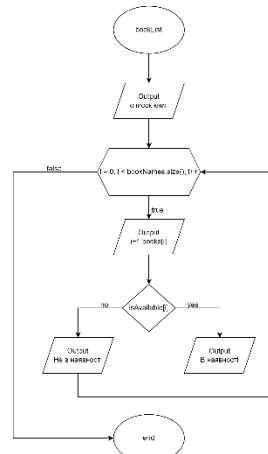
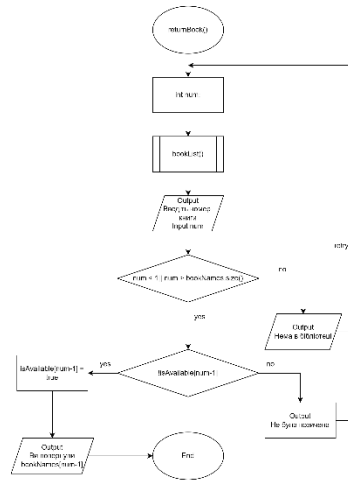
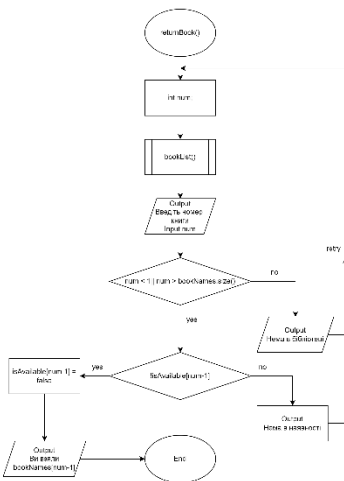
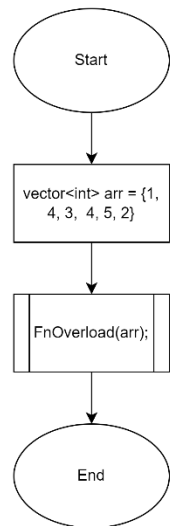
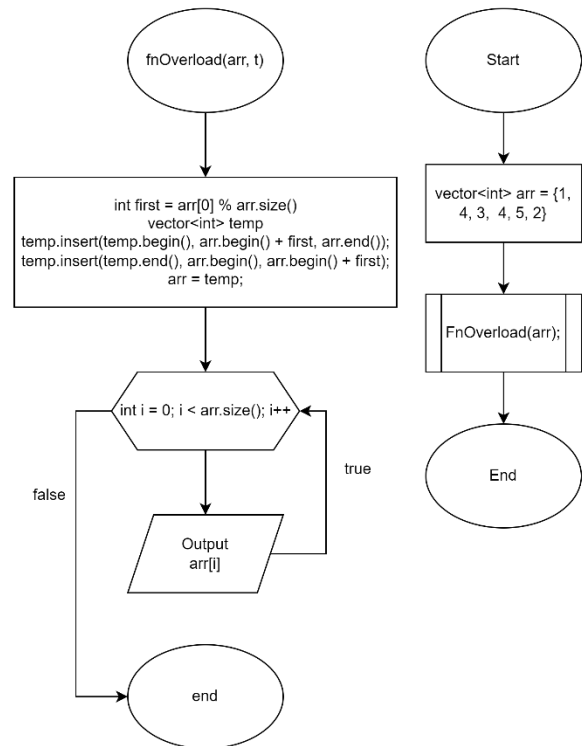
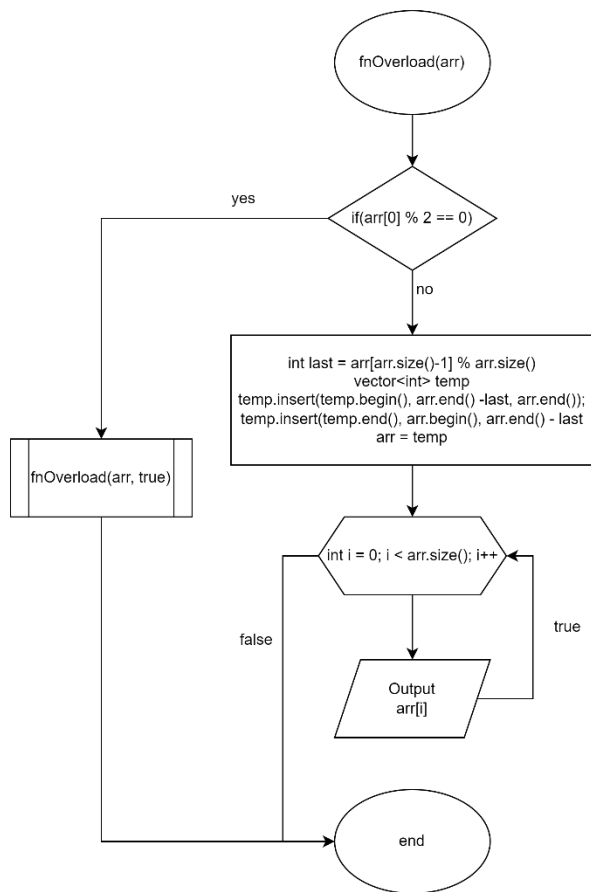
## **Виконання роботи**

### **Варіант 23**

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

**Time expected – 2 hours**

**Time spent – 2.5 hours**



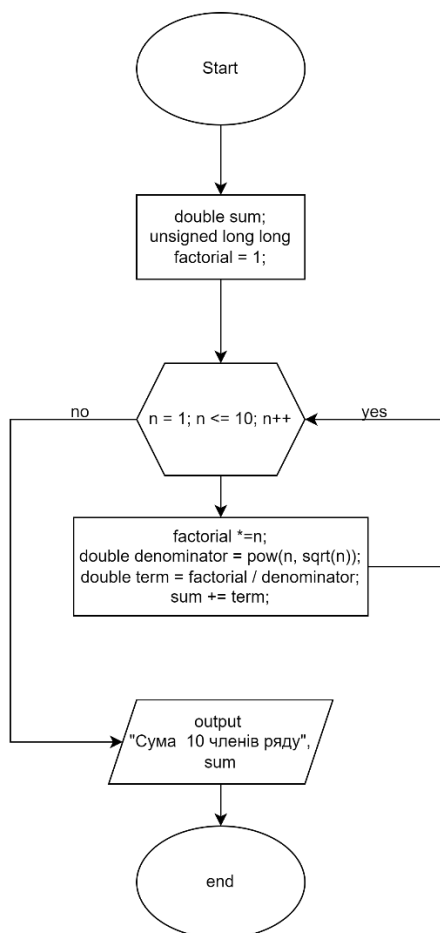
**Task 3 - Lab# programming: VNS Lab 2**  
 time expected – 30 m  
 time spent – 25m

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6
7  int main(){
8
9      double sum = 0.0;
10     unsigned long long factorial = 1;
11     for(int n = 1; n <= 10; n++){
12         factorial *= n;
13         double denominator = pow(n, sqrt(n));
14         double term = factorial / denominator;
15         sum += term;
16     }
17     cout << "Сума перших 10 членів ряду:  " << sum;
18
19 }

```

Сума перших 10 членів ряду: 3153.32



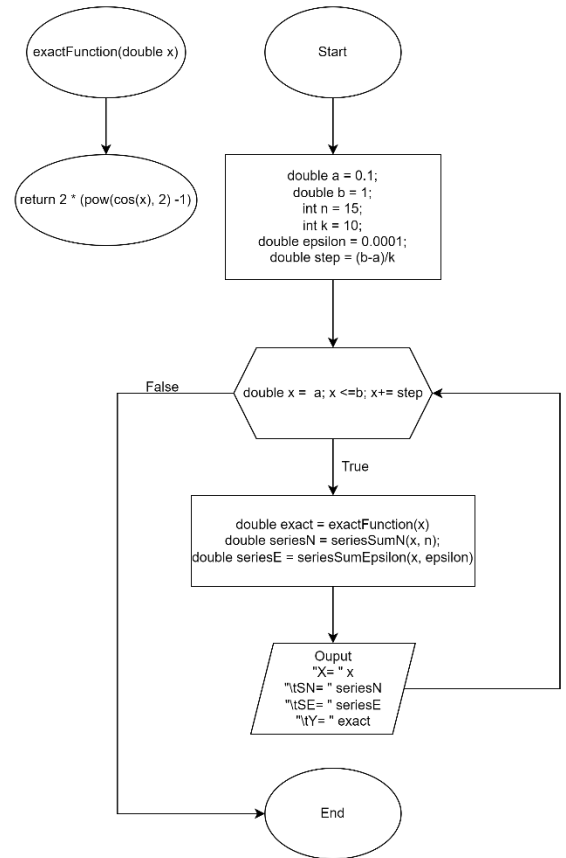
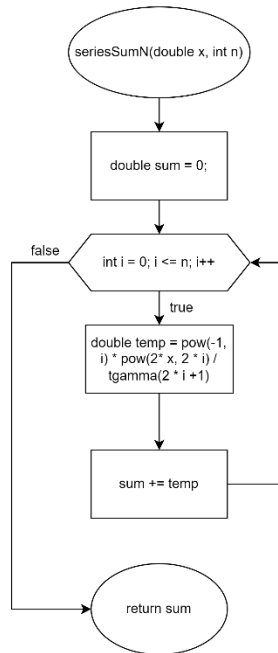
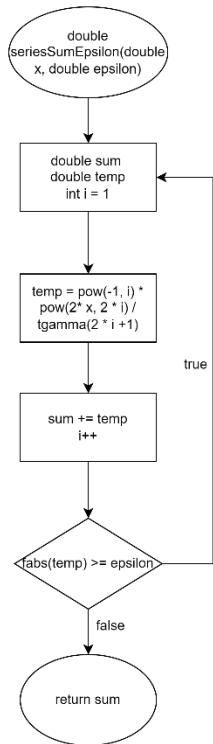
## Task 4 - Lab# programming: VNS Lab 3

time expected – 30 m

time spent – 50 m

```
1  #include <iostream>
2  #include <cmath>
3  #include <iomanip>
4
5  using namespace std;
6
7  double exactFunction(double x);
8  double seriesSumN(double x, int n);
9  double seriesSumEpsilon(double x, double epsilon);
10
11 int main() {
12     double a = 0.1;
13     double b = 1.0;
14     int n = 15;
15     int k = 10;
16     double epsilon = 0.0001;
17     double step = (b - a) / k;
18     cout << fixed << setprecision(6);
19
20     for (double x = a; x <= b; x += step) {
21         double exact = exactFunction(x);
22         double seriesN = seriesSumN(x, n);
23         double seriesE = seriesSumEpsilon(x, epsilon);
24
25         cout << "X= " << x << "\tSN= " << seriesN << "\tSE= " << seriesE << "\tY= " << exact << endl;
26     }
27
28     return 0;
29 }
30
31 double exactFunction(double x) {
32     return 2 * (pow(cos(x), 2) - 1);
33 }
34
35 double seriesSumN(double x, int n) {
36     double sum = 0.0;
37     for (int i = 0; i <= n; i++) {
38         double temp = pow(-1, i) * pow(2 * x, 2 * i) / tgamma(2 * i + 1);
39         sum += temp;
40     }
41     return sum;
42 }
43
44 double seriesSumEpsilon(double x, double epsilon) {
45     double sum = 0.0;
46     double temp;
47     int i = 1;
48     do {
49         temp = pow(-1, i) * pow(2 * x, 2 * i) / tgamma(2 * i + 1);
50         sum += temp;
51         i++;
52     } while (fabs(temp) >= epsilon);
53     return sum;
54 }
```

X= 0.460000	SN= 0.605820	SE= -0.394180	Y= -0.394180
X= 0.550000	SN= 0.453596	SE= -0.546403	Y= -0.546404
X= 0.640000	SN= 0.286715	SE= -0.713285	Y= -0.713285
X= 0.730000	SN= 0.110570	SE= -0.889430	Y= -0.889430
X= 0.820000	SN= -0.069148	SE= -1.069149	Y= -1.069148
X= 0.910000	SN= -0.246632	SE= -1.246632	Y= -1.246632
X= 1.000000	SN= -0.416147	SE= -1.416147	Y= -1.416147



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

**Time expected – 1.5 hour**

**Time spent – 2.5 hours**

1)

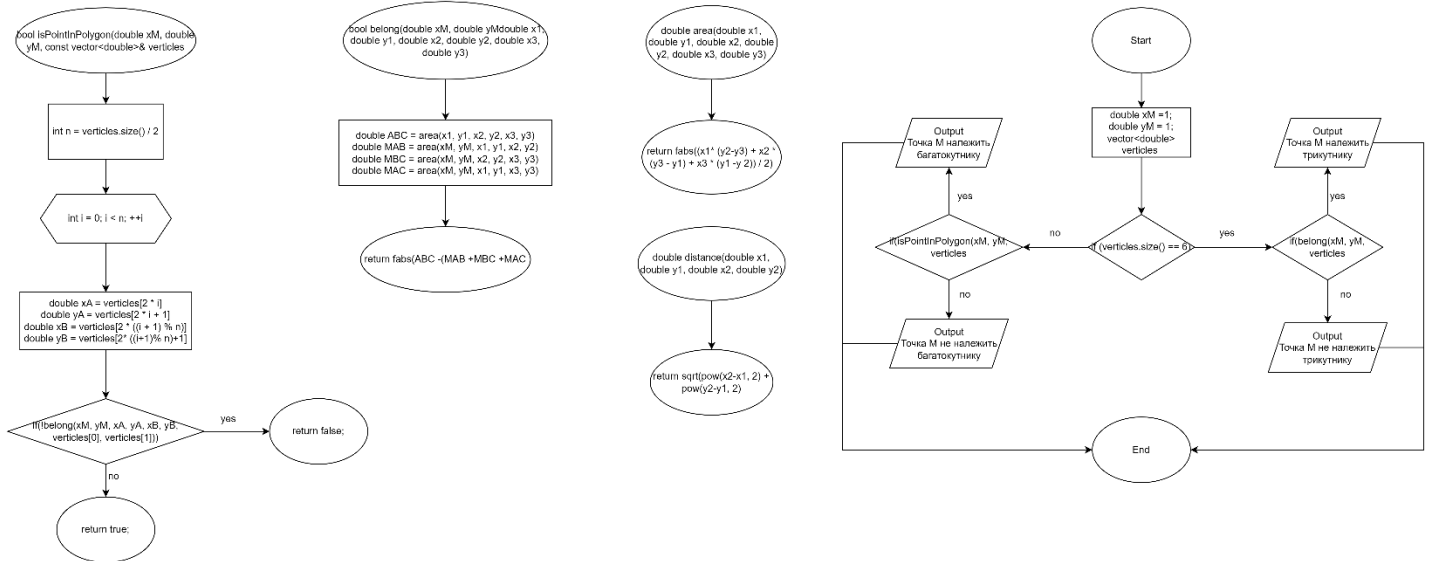
```
1  #include <iostream>
2  #include <cmath>
3  #include <vector>
4
5  using namespace std;
6
7  double distance(double x1, double y1, double x2, double y2);
8  double area(double x1, double y1, double x2, double y2, double x3, double y3);
9  bool belong(double xM, double yM, double x1, double y1, double x2, double y2, double x3, double y3);
10 bool isPointInPolygon(double xM, double yM, const vector<double>& vertices);
11
12 int main() {
13     double xM = 1;
14     double yM = 1;
15     vector<double> vertices = {0, 0, 1, 1.5, 2, 0, 4, 2.2};
16
17     if (vertices.size() == 6) {
18         if (belong(xM, yM, vertices[0], vertices[1], vertices[2], vertices[3], vertices[4], vertices[5])) {
19             cout << "Точка М належить трикутнику." << endl;
20         } else {
21             cout << "Точка М не належить трикутнику." << endl;
22         }
23     } else {
24         if (isPointInPolygon(xM, yM, vertices)) {
25             cout << "Точка М належить багатокутнику." << endl;
26         } else {
27             cout << "Точка М не належить багатокутнику." << endl;
28         }
29     }
30
31     return 0;
32 }
33
34 double distance(double x1, double y1, double x2, double y2){
35     return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));
36 }
37
38 double area(double x1, double y1, double x2, double y2, double x3, double y3){
39     return fabs((x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2)) / 2.0);
40 }
41
42 bool belong(double xM, double yM, double x1, double y1, double x2, double y2, double x3, double y3){
43     double ABC = area(x1, y1, x2, y2, x3, y3);
44     double MAB = area(xM, yM, x1, y1, x2, y2);
45     double MBC = area(xM, yM, x2, y2, x3, y3);
46     double MAC = area(xM, yM, x1, y1, x3, y3);
47
48     return fabs(ABC - (MAB + MBC + MAC));
49 }
50
51 bool isPointInPolygon(double xM, double yM, const vector<double>& vertices){
52     int n = vertices.size() / 2;
53     for (int i = 0; i < n; ++i) {
54         double xA = vertices[2 * i];
55         double yA = vertices[2 * i + 1];
56         double xB = vertices[2 * ((i + 1) % n)];
57         double yB = vertices[2 * ((i + 1) % n) + 1];
58         if (!belong(xM, yM, xA, yA, xB, yB, vertices[0], vertices[1])) {
59             return false;
60         }
61     }
62     return true;
63 }
```

```
double xM = 1;
double yM = 1;
vector<double> vertices = {0, 0, 1, 1.5, 2, 0, 4, 2.2};
```

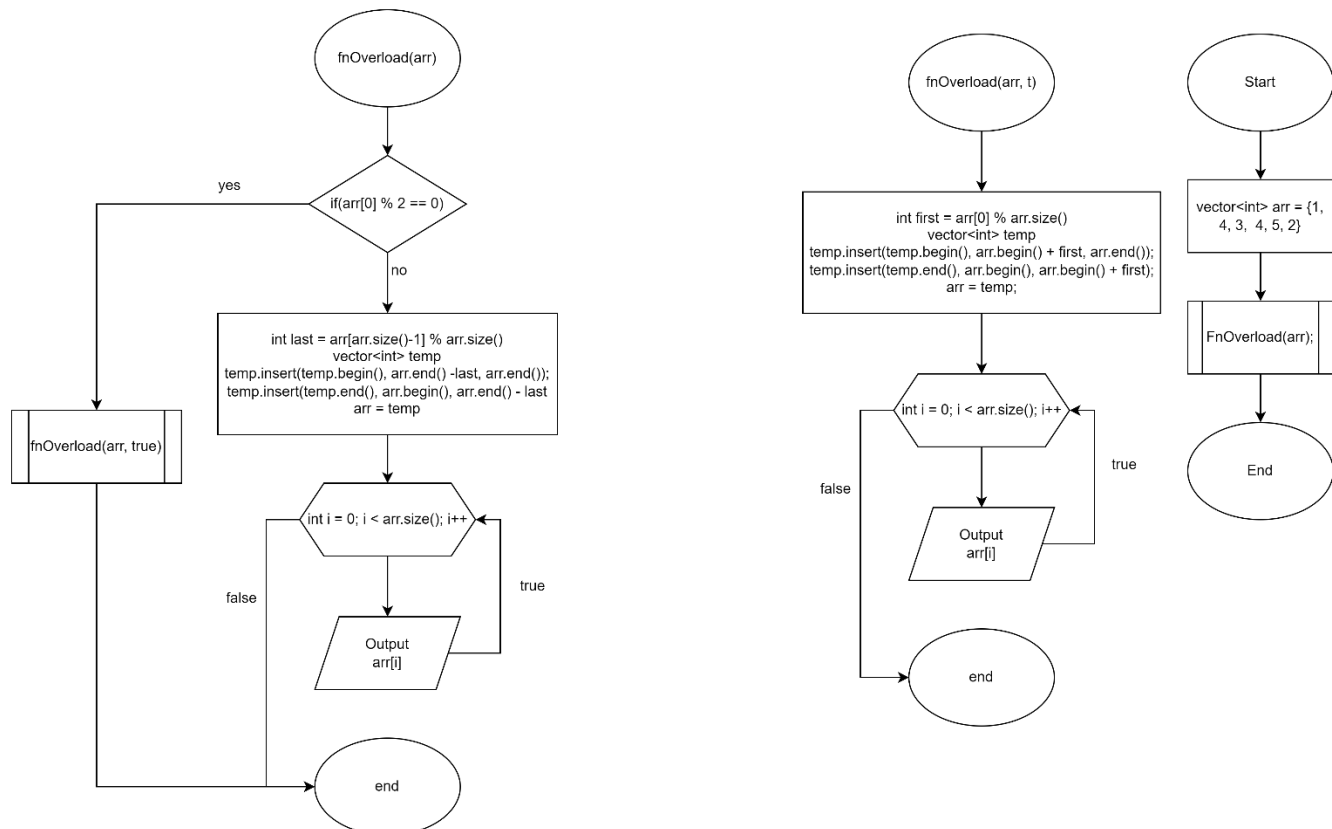
Точка М не належить багатокутнику.

```
double xM = 1;
double yM = 1;
vector<double> vertices = {0, 3, 1, 2, 5, 2.2};
```

Точка М належить трикутнику.



2)





```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  void FnOverload(vector<int> &arr, bool t)
6  {
7      int first = arr[0] % arr.size();
8      vector<int> temp;
9      temp.insert(temp.begin(), arr.begin() + first, arr.end());
10     temp.insert(temp.end(), arr.begin(), arr.begin() + first);
11     arr = temp;
12     for (int i = 0; i < arr.size(); i++)
13     {
14         cout << arr[i] << " ";
15     }
16     cout << endl;
17 }
18
19 void FnOverload(vector<int> &arr)
20 {
21     if (arr[0] % 2 == 0)
22         FnOverload(arr, true);
23     else
24     {
25         int last = arr[arr.size() - 1] % arr.size();
26         vector<int> temp;
27         temp.insert(temp.begin(), arr.end() - last, arr.end());
28         temp.insert(temp.end(), arr.begin(), arr.end() - last);
29         arr = temp;
30         for (int i = 0; i < arr.size(); i++)
31         {
32             cout << arr[i] << " ";
33         }
34         cout << endl;
35     }
36 }
37
38 int main()
39 {
40     vector<int> arr = {1, 4, 3, 4, 5, 2};
41     FnOverload(arr);
42
43     return 0;
44 }

```

```
vector<int> arr = {2, 4, 3, 4, 5, 2}; 3 4 5 2 2 4
```

```
vector<int> arr = {3, 4, 3, 4, 5, 2}; 5 2 3 4 3 4
```

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

time expected – 1.5 hour

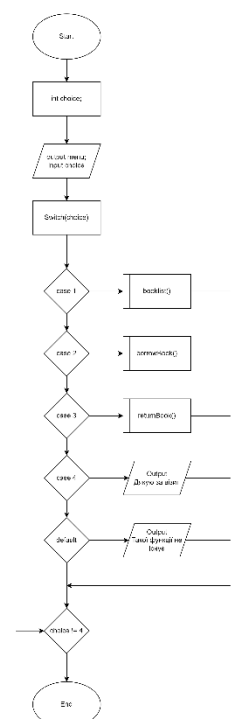
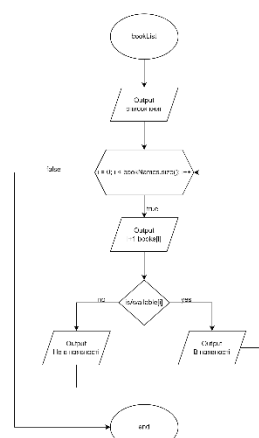
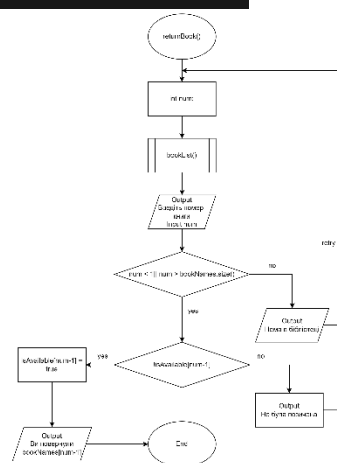
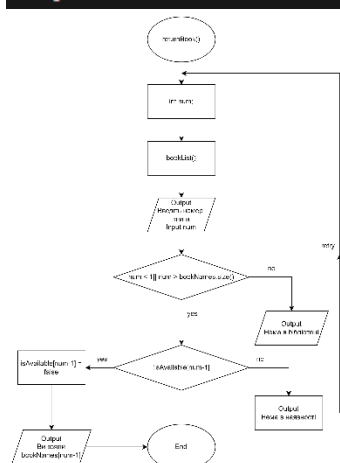
Time spent – 1.5 hour

```
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 void bookslist();
7 void borrowBook();
8 void returnBook();
9
10 vector<string> bookNames = {"Bible", "Harry Potter", "Sentimental story", "Bersek", "Vinland saga"};
11 vector<bool> isAvailable = {true, true, true, true, false};
12
13 int main()
14 {
15     int choice;
16     do{
17         cout << "*****" << endl;
18         cout << "Меню:" << endl;
19         cout << "1. Переглянути [у] книги:" << endl;
20         cout << "2. Позичити книгу" << endl;
21         cout << "3. Повернути книгу" << endl;
22         cout << "4. Вийти" << endl;
23         cout << "*****" << endl;
24         cout << "Введіть свій вибір: ";
25         cin >> choice;
26
27         switch(choice){
28             case 1:
29                 bookslist();
30                 break;
31             case 2:
32                 borrowBook();
33                 break;
34             case 3:
35                 returnBook();
36                 break;
37             case 4:
38                 cout << "Дякуємо за візит";
39                 break;
40             default:
41                 cout << "Такої функції не існує" << endl;
42         }
43     }while(choice != 4);
44 }
45
46 void bookslist(){
47     cout << "Список книг: " << endl;
48     for(int i = 0; i < bookNames.size(); i++){
49         cout << i+1 << " " << bookNames[i];
50         if(isAvailable[i]){
51             cout << " [ ] наявності" << endl;
52         }else{
53             cout << " [X] в наявності" << endl;
54         }
55     }
56 }
57
58 };
59
60 void borrowBook(){
61     int num;
62     retry:
63     bookslist();
64     cout << "Введіть номер книги, яку хочете взяти: " << endl;
65     cin >> num;
66     if(num < 1 || num > bookNames.size()){
67         cout << "Цієї книги немає в бібліотеці" << endl;
68         goto retry;
69     }
70     if(isAvailable[num - 1]){
71         isAvailable[num - 1] = false;
72         cout << "Ви взяли книгу " << bookNames[num - 1] << endl;
73     }else{
74         cout << "Книги немає в наявності" << endl;
75         goto retry;
76     }
77 }
78
79 };
80
81 void returnBook(){
82     int num;
83     retry:
84     bookslist();
85     cout << "Введіть номер книги, яку хочете повернути: " << endl;
86     cin >> num;
87     if(num < 1 || num > bookNames.size()){
88         cout << "Цієї книги немає в бібліотеці" << endl;
89         goto retry;
90     }
91     if(!isAvailable[num - 1]){
92         isAvailable[num - 1] = true;
93         cout << "Ви повернули книгу " << bookNames[num - 1] << endl;
94     }else{
95         cout << "Книга не була позичена " << endl;
96         goto retry;
97     }
98 }
99
100 };
```

```

*****
Меню:
1. Переглянути усі книги:
2. Позичити книгу
3. Повернути книгу
4. Вийти
*****
Введіть свій вибір: 2
Список книг:
1 Bible В наявності
2 Harry Potter В наявності
3 Sentimental story В наявності
4 Berserk В наявності
5 Vinland saga Не в наявності
Введіть номер книги, яку хочете взяти:
7
Цієї книги немає в бібліотеці
Список книг:
1 Bible В наявності
2 Harry Potter В наявності
3 Sentimental story В наявності
4 Berserk В наявності
5 Vinland saga Не в наявності
Введіть номер книги, яку хочете взяти:
3
Ви взяли книгу Sentimental story
*****
Меню:
1. Переглянути усі книги:
2. Позичити книгу
3. Повернути книгу
4. Вийти
*****
Введіть свій вибір: 4
Дякуємо за візит

```



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

Time expected – 30m

Time spent – 15m

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int n;
7      cin >> n;
8
9      int supermanWins = 0;
10     int batmanWins = 0;
11
12     for(int i = 0; i < n; i++){
13         int a, b, c;
14         cin >> a >> b >> c;
15
16         int nim_sum = a ^ b ^ c;
17
18         if(nim_sum != 0){
19             supermanWins++;
20         }else{
21             batmanWins++;
22         }
23     }
24     cout << supermanWins << "-" << batmanWins;
25
26
27     return 0;
28 }
```

2	2
4 5 19	3 4 5
2 3 4	7 0 7
2-0	1-1

Problem

Successful solutions

Html

Pdf

Submit a solution

## Супермен vs. Бетмен

Limits: 2 sec., 256 MiB

Одного прекрасного зимового вечора два супергерої Супермен та Бетмен, доскочу налітавшись львівським безкраїм небом, влаштували таку гру.

Є три цілих невід’ємних числа  $a$ ,  $b$ ,  $c$ . Двоє гравців по черзі роблять ходи. За одних хід гравець має закреслити одне з трьох чисел і на його місце записати строго менше ціле невід’ємне число. Очевидно, що закреслювати 0 не можна. Гравець, який не може зробити хід, програє.

У кожній партії першим завжди робить хід Супермен. Гра виявилася дуже цікавою, тому було зіграно не одну, а аж  $n$  партій.

Подивитися на такий турнір між двома улюбленицями зібралися майже всі діти Львова.

Ваше ж завдання — перерахувати умови гри ще декілька разів та визначити остаточний рахунок (за партіями) при оптимальній стратегії обох гравців.

### Input

У першому рядку міститься ціле число  $n$  — кількість зіграних партій.

Кожен із наступних  $n$  рядків містить трійку цілих чисел  $a$ ,  $b$ ,  $c$ .

### Output

У єдиному рядку виведіть рахунок у вигляді  $s-b$ , де  $s$  — кількість партій, які виграв Супермен, а  $b$  — кількість партій, які виграв Бетмен.

### Constraints

$1 \leq n \leq 100$ ,  
 $0 \leq a, b, c \leq 10^3$ .

### Samples

Input (stdin)	Output (stdout)
2	1 -1

Submit a solution

File

Source Code

Compiler

C++ 23

File

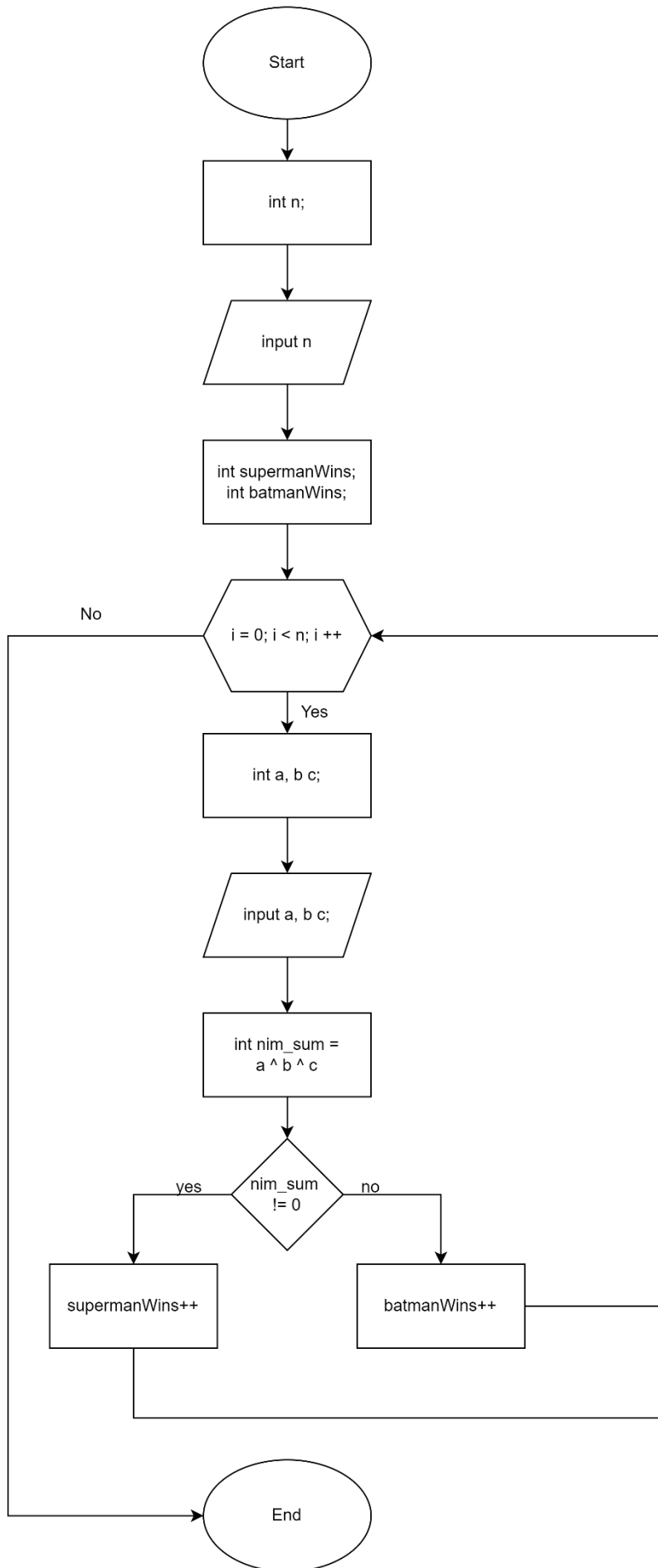
Вибрати файл

Файл не вибрано

Submit

↻

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
4 hours ago	C++ 23	Accepted	0.003	1.039	<a href="#">View</a>



## Task 8 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

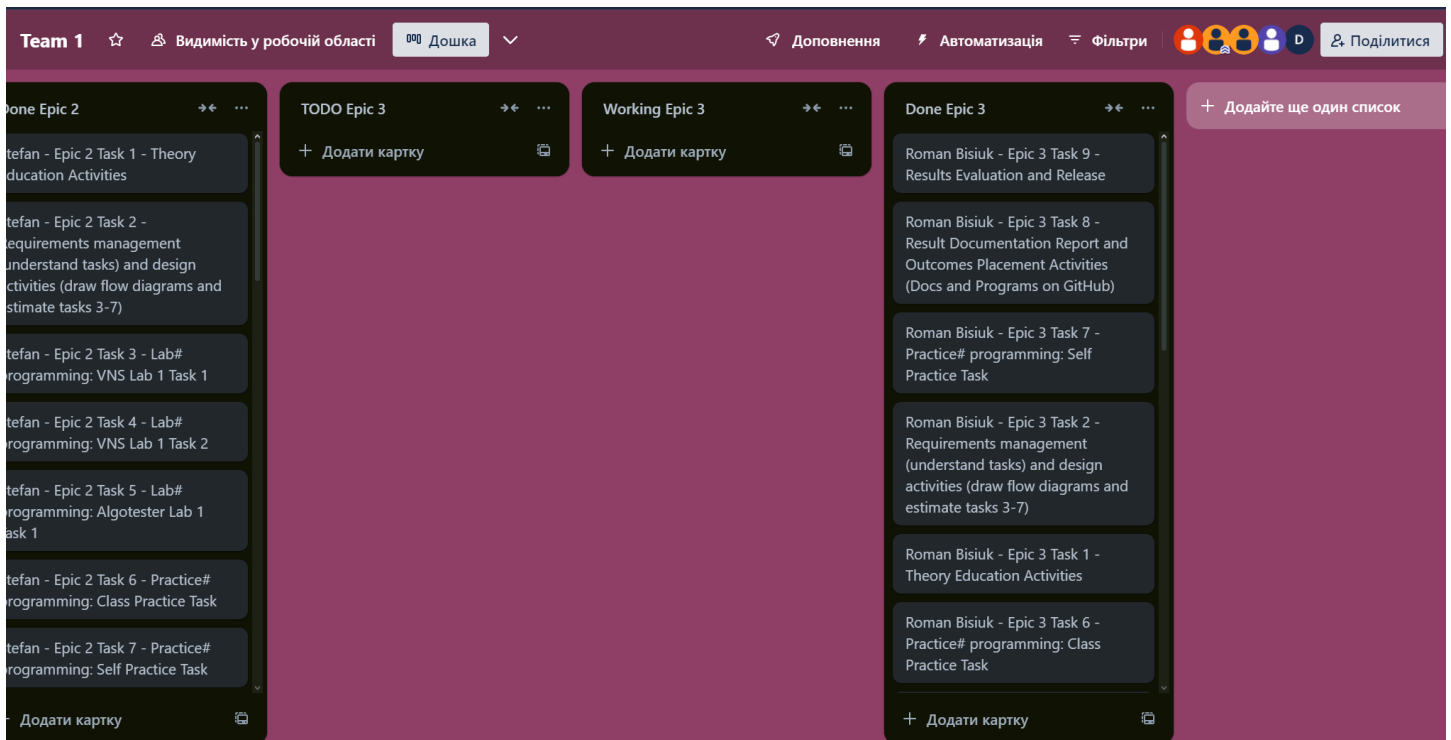
Time expected – 1 hour

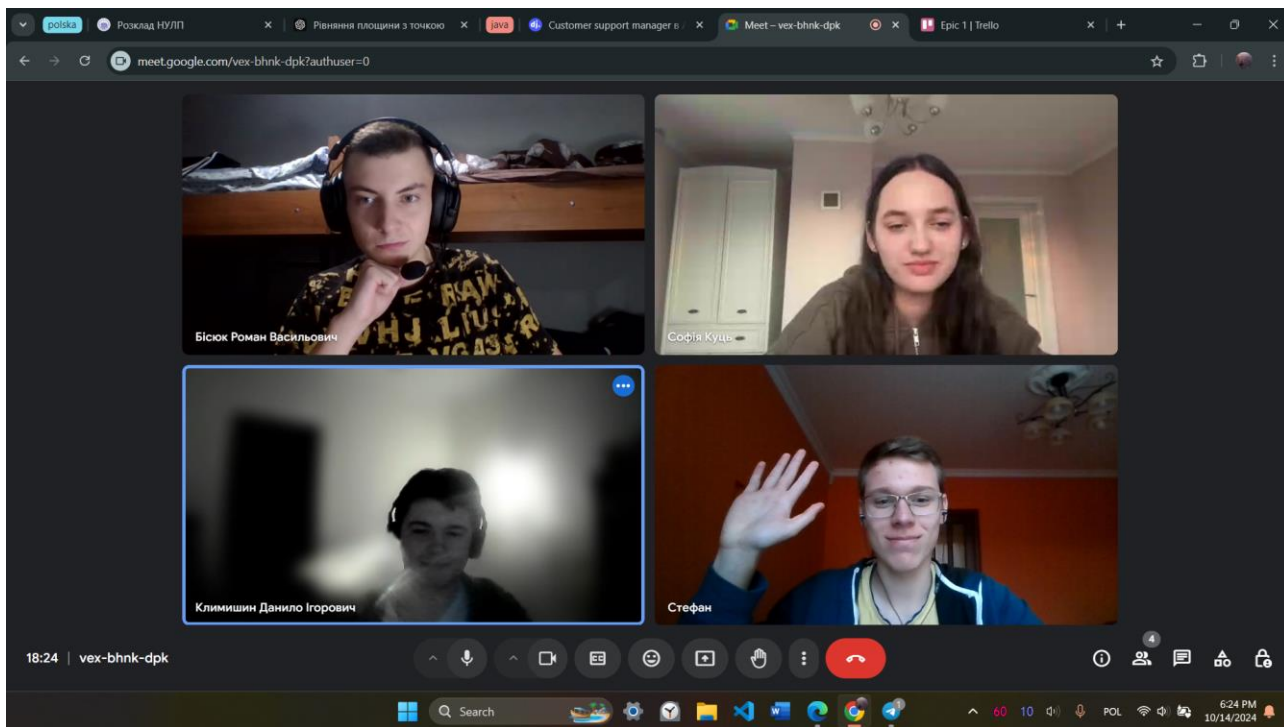
Spent – ~1 hour

## Task 9 - Results Evaluation and Release

### Pull-Request:

**Meets:** розібралися з дошкою в Trello, допомогли один одному з написанням практичного завдання та блок-схеми до нього





**Висновок:** За лабораторну роботу я навчився використовувати різні типи циклів і управляти їх виконанням, створювати функції з різними параметрами, застосовувати рекурсію, а також зрозумів принципи використання просторів імен і перевантаження функцій у програмуванні.