

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-12
Гаврих Юрій Дмитрович

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:

Дослідження та впровадження різних типів циклів, включаючи прості та вкладені цикли, для автоматизації повторюваних дій у програмах. Вивчення механізмів завершення виконання циклів, зокрема команд break та continue. Опанування функцій як основного інструменту для розбиття програми на логічні модулі, зокрема робота з передачею аргументів та поверненням значень. Перевантаження функцій, опанування функцій з змінною кількістю параметрів (еліпсис). Робота з рекурсією для вирішення складних завдань Застосування вбудованих функцій для виконання типових операцій, таких як математичні розрахунки або робота з рядками.

Теоретичні відомості:

[Цикли в C++](#)

[Функції в C++](#)

[Перевантаження функцій](#)

[Еліпсис](#)

Виконання роботи

Task 3 - Lab# programming: VNS Lab 2

Час виконання ~ 40 хв.

15) Знайти суму ряду з точністю $\epsilon=0.0001$, загальний член якого

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

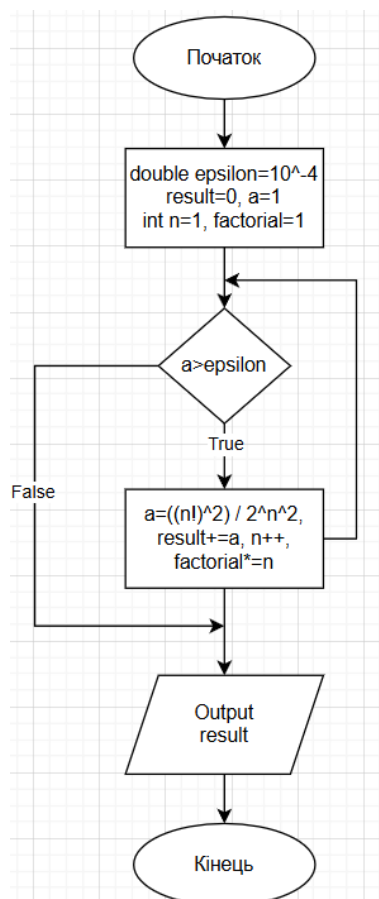
Розв'язок:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  int main(){
5      double epsilon=0.0001;
6      double result=0;
7      int n=1;
8      double a=1;
9      int factorial=1;
10     while(a>epsilon){
11         a= pow(factorial,2) / (pow(2,n*n));
12         result+=a;
13         n++;
14         factorial*=n;
15     }
16     printf("Sum= %.8lf",result);
17 }
```

Результат виконання:

Sum= 0.82953826

Блок-схема:



Task 4 - Lab# programming: VNS Lab 3

Час виконання ~ 1 год.

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.

15	$y = \frac{1+x^2}{2} \operatorname{arctg} x - \frac{x}{2}$	$0,1 \leq x \leq 1$	30	$S = \frac{x^3}{3} - \frac{x^5}{15} + \dots + (-1)^{n+1} \frac{x^{2n+1}}{4n^2 - 1}$
----	--	---------------------	----	---

Розв'язок:

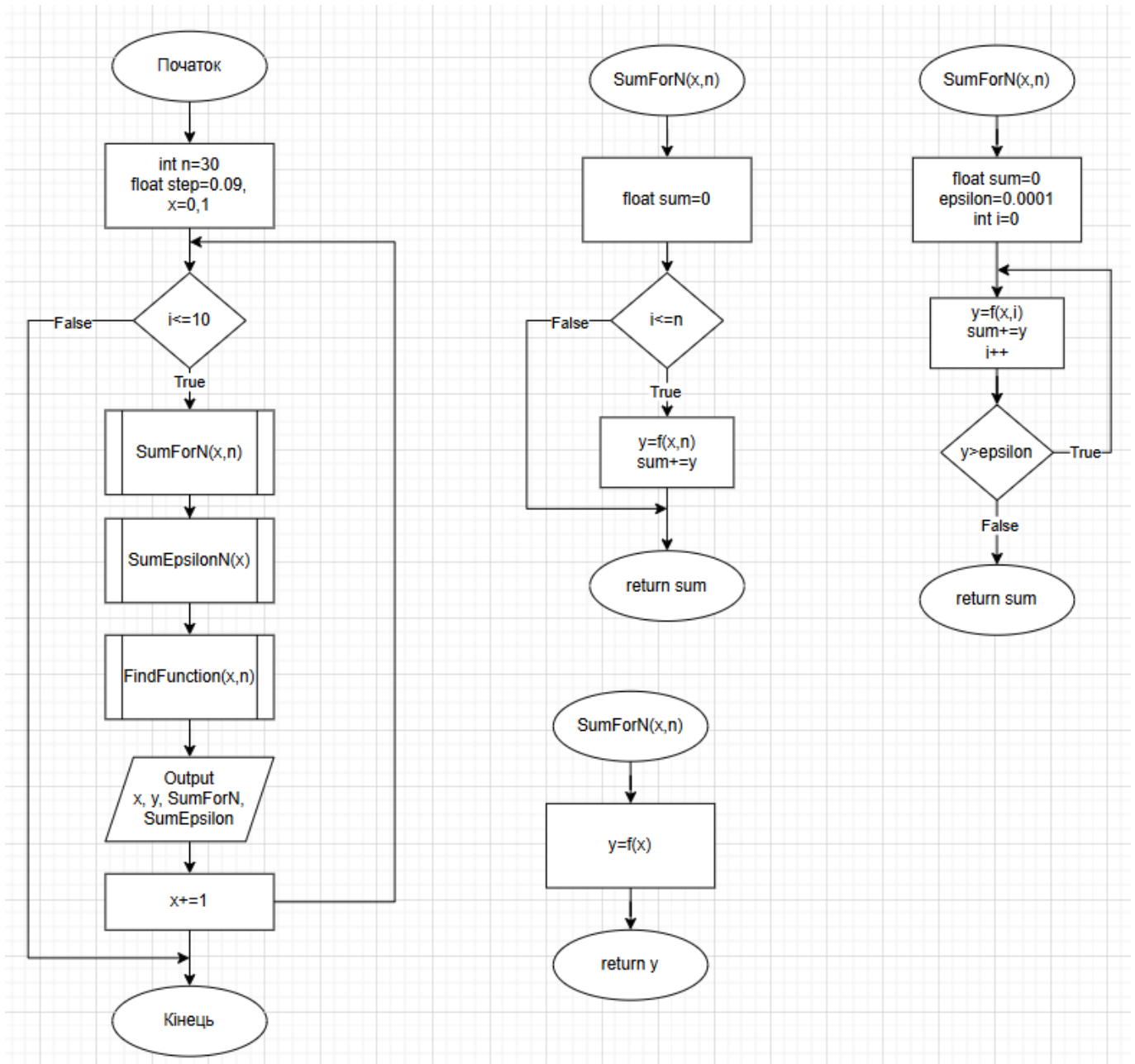
```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  float FindFunction(float x){
5      float y = (( 1 + (x*x) ) / 2) * atan(x) - ( x/2 );
6      return y;
7  }
8
9  float SumForN (float x, int n){
10     float sum=0;
11     for(int i=1; i<=n; i++){
12         float y= ( pow ( -1 , i + 1 ) ) * ( pow ( x , 2 * i + 1 ) / ( 4 * i * i - 1 ) );
13         sum+=y;
14     }
15     return sum;
16 }
17
18 float SumForEpsilon(float x){
19     float epsilon=0.0001;
20     float sum=0;
21     int i=1;
22     float y;
23     do{
24         y= ( pow ( -1 , i + 1 ) ) * ( pow ( x , 2 * i + 1 ) / ( 4 * i * i - 1 ) );
25         sum+=y;
26         i++;
27     }while(abs(y)>epsilon);
28     return sum;
29 }
30
31 int main(){
32
33     int n=30;
34     float step = 0.09; // a=0.1 b=1 k=10 --> (1-0.1)/10
35     float x=0.1; // a=0.1
36     for(int i=0; i<=10; i++){
37         float rezN = SumForN(x,n);
38         float rezEpsilon = SumForEpsilon(x);
39         float y=FindFunction(x);
40         printf("x=      %.2f\n",x);
41         printf("f(x)=    %f\n",y);
42         printf("Sum_n=     %f\n",rezN);
43         printf("Sum_eps=   %f\n",rezEpsilon);
44         printf("-----\n");
45         x+=step;
46     }
47 }
```

Результат виконання:

```
x=      0.10
f(x)=    0.000333
Sum_n=    0.000333
Sum_eps=  0.000333
-----
x=      0.19
f(x)=    0.002270
Sum_n=    0.002270
Sum_eps=  0.002270
-----
x=      0.28
f(x)=    0.007206
Sum_n=    0.007206
Sum_eps=  0.007206
-----
x=      0.37
f(x)=    0.016447
Sum_n=    0.016447
Sum_eps=  0.016449
-----
```

```
x=      0.46
f(x)=    0.031184
Sum_n=    0.031184
Sum_eps=  0.031182
-----
x=      0.55
f(x)=    0.052477
Sum_n=    0.052477
Sum_eps=  0.052465
-----
x=      0.64
f(x)=    0.081252
Sum_n=    0.081252
Sum_eps=  0.081268
-----
x=      0.73
f(x)=    0.118306
Sum_n=    0.118306
Sum_eps=  0.118319
-----
x=      0.82
f(x)=    0.164317
Sum_n=    0.164317
Sum_eps=  0.164342
-----
x=      0.91
f(x)=    0.219855
Sum_n=    0.219854
Sum_eps=  0.219820
-----
x=      1.00
f(x)=    0.285398
Sum_n=    0.285264
Sum_eps=  0.285445
-----
```

Блок-схема:



Task 5 - Lab# programming: VNS Lab 7

Час виконання ~ 2 год.

Частина 1

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

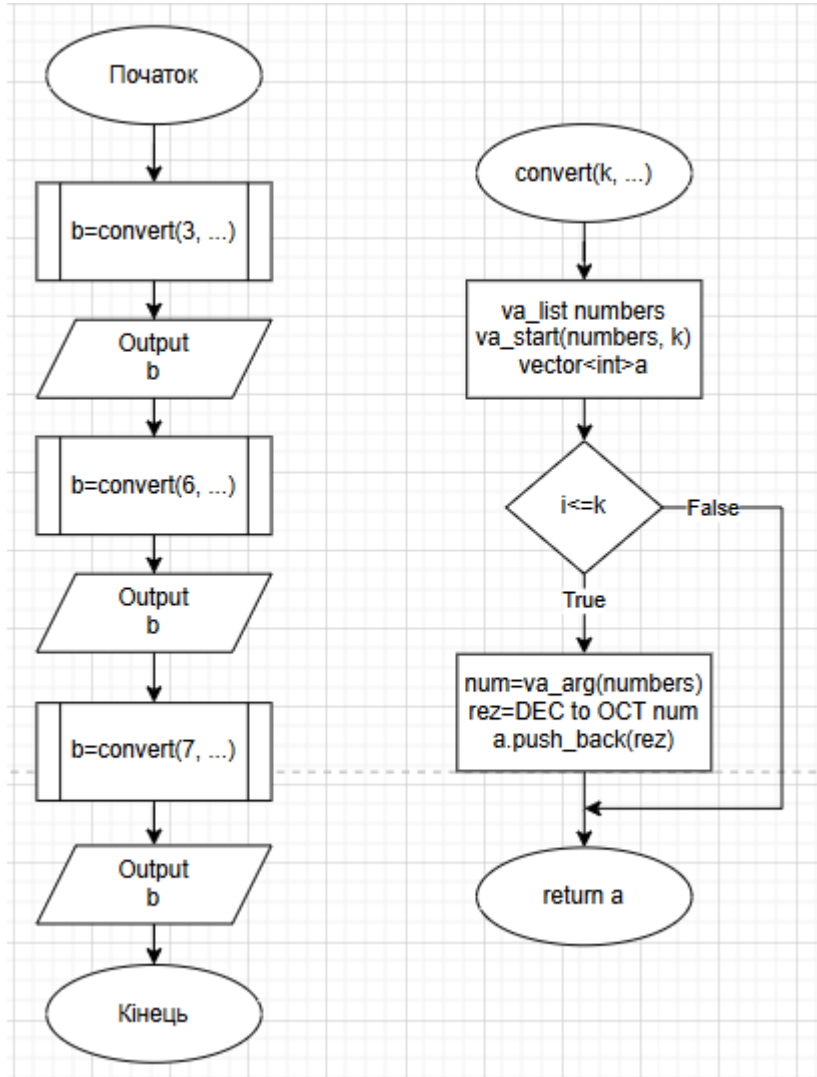
Розв'язок:

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  vector<int> convert(int k, ...) {
6      va_list numbers;
7      va_start(numbers, k);
8      vector<int> a;
9      for (int i = 0; i < k; i++) {
10         int num = va_arg(numbers, int);
11         string s = to_string(num);
12         reverse(s.begin(), s.end());
13         int cur = 0;
14         int rez = 0;
15         for (auto j: s) {
16             rez += (j - '0') * pow(8, cur);
17             cur++;
18         }
19         a.push_back(rez);
20     }
21     va_end(numbers);
22     return a;
23 }
24
25
26 int main() {
27     vector<int> b = convert(3, 7, 11, 34);
28     for (int i: b) {
29         cout << i << " ";
30     }
31     cout << endl;
32
33     b = convert(6, 4, 12, 13, 15, 16, 17);
34     for (int i: b) {
35         cout << i << " ";
36     }
37     cout << endl;
38
39     b = convert(7, 20, 21, 22, 23, 24, 25, 26);
40     for (int i: b) {
41         cout << i << " ";
42     }
43     cout << endl;
44 }
```

Результат виконання:

```
7 9 28
4 10 11 13 14 15
16 17 18 19 20 21 22
```

Блок-схема:



Частина 2

Написати перевантажені функції й основну програму, що їх викликає.

- а) цілі числа підносить до степеня n ;
- б) з десяткових чисел бере корінь степеня n .

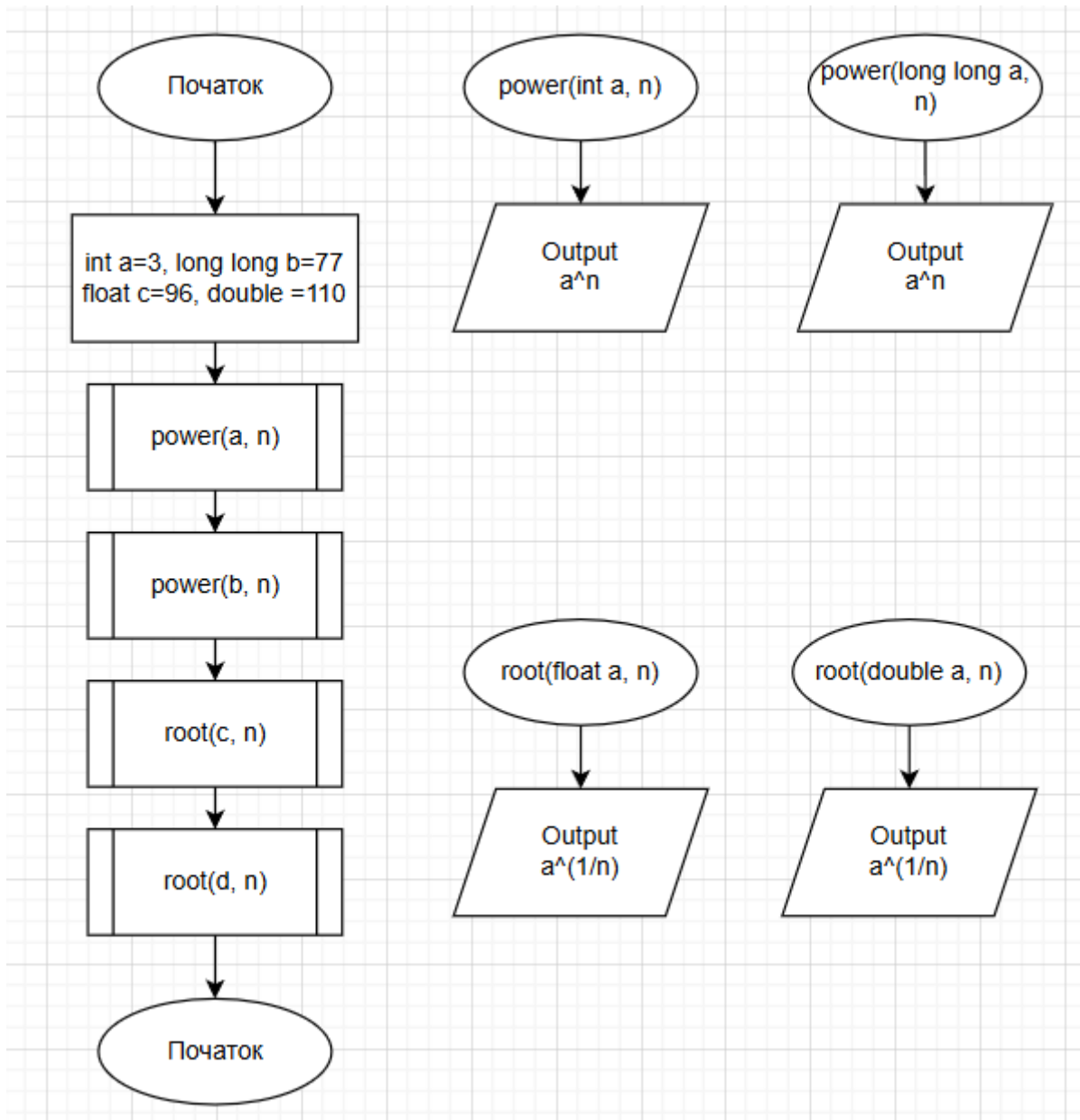
Розв'язок:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  void power(int a, int n){
5      cout<<"int power "<<pow(a,n)<<endl;
6  }
7
8  void power(long long a, int n){
9      cout<<"long long power "<<pow(a,n)<<endl;
10 }
11
12 void root(float a, int n){
13     cout<<"float root "<<pow(a,1.0/n)<<endl;
14 }
15
16 void root(double a, int n){
17     cout<<"double root "<<pow(a,1.0/n)<<endl;
18 }
19
20 int main(){
21     int a=3;
22     long long b=77;
23     float c=96;
24     double d=110;
25     power(a,2);
26     power(b,2);
27     root(c,2);
28     root(d,2);
29 }
```

Результат виконання:

```
int power 9
long long power 5929
float root 9.79796
double root 10.4881
```

Блок-схема:



Task 6 - Practice# programming: Class Practice Task

Час виконання ~ 1,5 год.

Менеджмент бібліотеки

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.

Розв'язок:

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  struct Book {
6      string name;
7      bool available;
8  };
9
10 int main() {
11     vector<Book> books = {"Book1", true},
12                          {"Book2", true},
13                          {"Book3", true},
14                          {"Book4", true};
15
16     int choice;
17     MENU:
18     cout << endl << "Меню:" << endl;
19     cout << "1. Список всіх книг" << endl;
20     cout << "2. Взяти книгу" << endl;
21     cout << "3. Повернути книгу" << endl;
22     cout << "4. Вийти" << endl;
23     cout << "Оберіть дію: ";
24     cin >> choice;
25     switch (choice) {
26     case 1: {
27         for (Book book: books) {
28             cout << book.name << " ";
29             if (book.available) cout << "Доступна" << endl;
30             else cout << "Недоступна" << endl;
31         }
32         break;
33     case 2: {
34         int book_num;
35         cout << "Введіть номер книги ";
36         cin >> book_num;
37         if ((book_num < 1) || (book_num > books.size())) {
38             cout << "Книги з таким номером не існує" << endl;
39             goto MENU;
40         } else if (!books[book_num - 1].available) {
41             cout << "Ця книга недоступна" << endl;
42             goto MENU;
43         } else {
44             cout << "Ви взяли книгу №" << book_num << " " << books[book_num - 1].name << endl;
45             books[book_num - 1].available = false;
46         }
47         break;
48     case 3: {
49         int book_num;
50         cout << "Введіть номер книги ";
51         cin >> book_num;
52         if ((book_num < 1) || (book_num > books.size())) {
53             cout << "Книги з таким номером не існує" << endl;
54             goto MENU;
55         } else if (books[book_num - 1].available) {
56             cout << "Ця книга не була позичена" << endl;
57             goto MENU;
58         } else {
59             cout << "Ви повернули книгу №" << book_num << " " << books[book_num - 1].name << endl;
60             books[book_num - 1].available = true;
61         }
62         break;
63     case 4: {
64         cout << "Гарного дня!" << endl;
65         exit(0);
66     }
67     default: {
68         cout << "Невідома дія, спробуйте ще раз";
69     }
70 }
71
72 cout << "Бажаєте виконати іншу операцію? (1 -> так, 0 -> ні)" << endl;
73 cin >> choice;
74 if (choice) goto MENU;
75 else cout << "Гарного дня!" << endl;
76 }
```

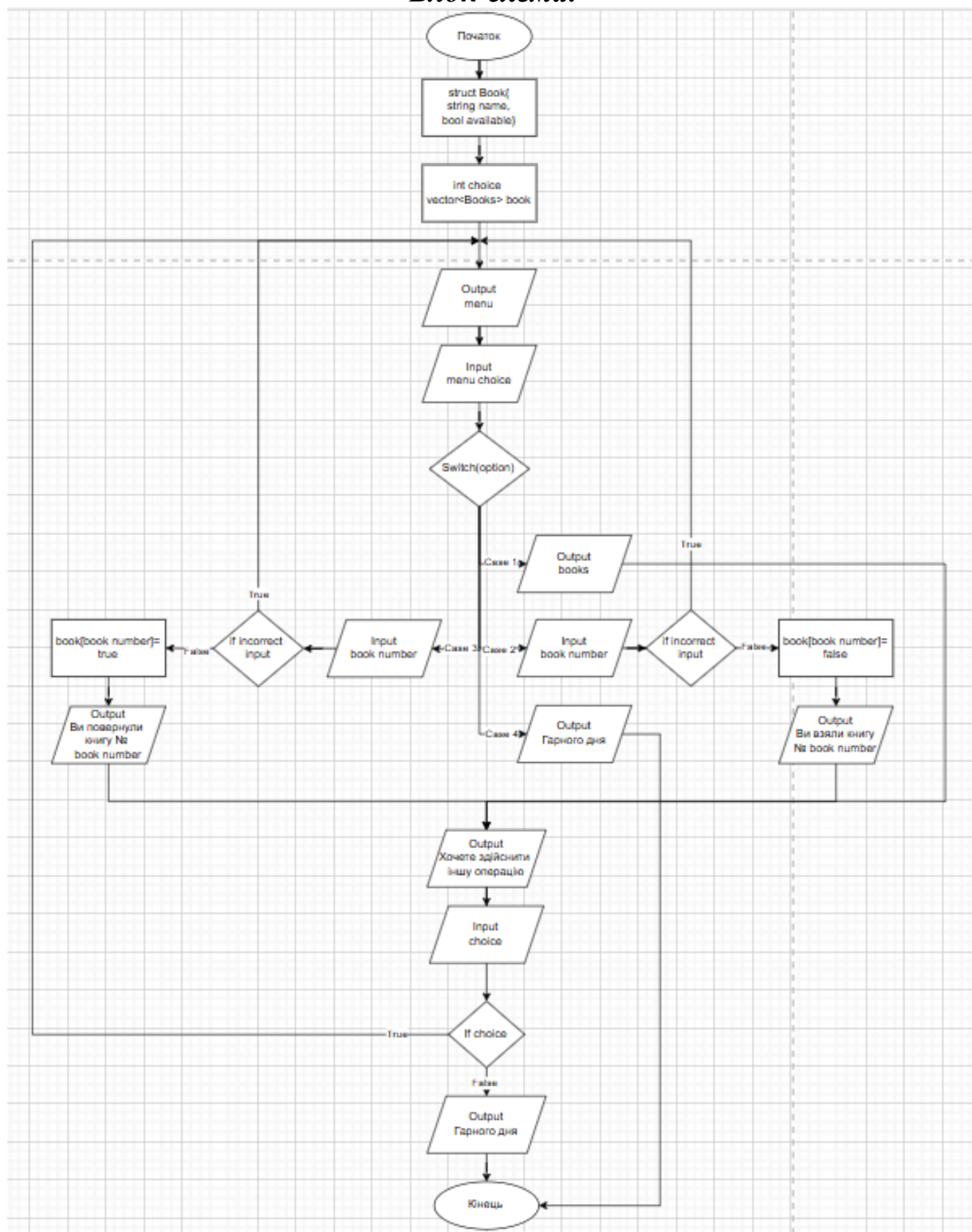
Результат виконання:

```
Меню:
1. Список всіх книг
2. Взяти книгу
3. Повернути книгу
4. Вийти
Оберіть дію: 2
Введіть номер книги 2
Ви взяли книгу №2 Book2
Бажаєте виконати іншу операцію? (1 -> так, 0 -> ні)
1

Меню:
1. Список всіх книг
2. Взяти книгу
3. Повернути книгу
4. Вийти
Оберіть дію: 3
Введіть номер книги 2
Ви повернули книгу №2 Book2
Бажаєте виконати іншу операцію? (1 -> так, 0 -> ні)
1

Меню:
1. Список всіх книг
2. Взяти книгу
3. Повернути книгу
4. Вийти
Оберіть дію: 4
Гарного дня!
```

Блок-схема:



Task 7 - Practice# programming: Self Practice Task

Час виконання~ 5 год.

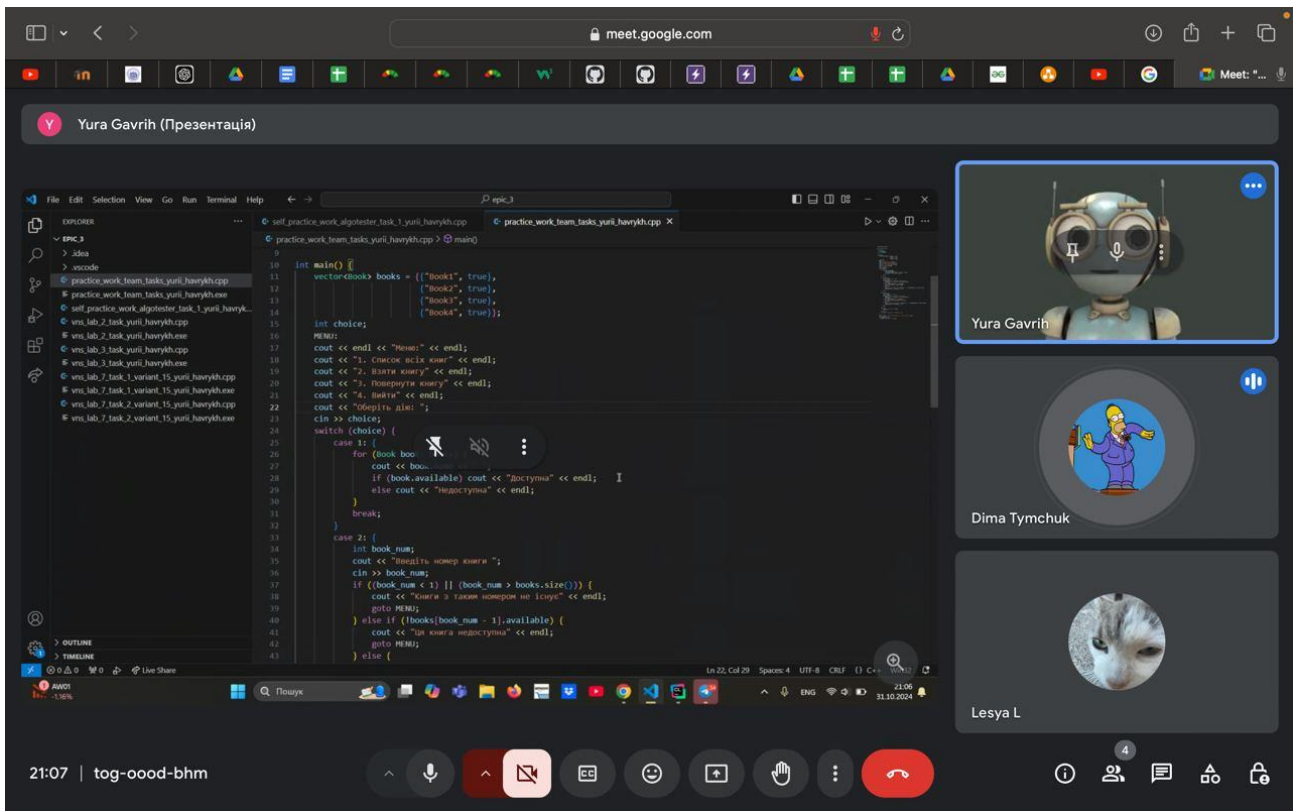
Algotester №2005 Гербарій

Розв'язок:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  const int N=200001;
5  int tin[N],tout[N],color[N];
6  vector<bool> visited(N,false);
7  vector<bool> visited1(N,false);
8  vector<vector<int>> graph(N);
9  vector<vector<int>> vertex_color(N);
10 vector<vector<int>> vertex_color2(N);
11 int timee=0;
12
13 void dfs1(int v){
14     timee++;
15     visited[v]=true;
16     tin[v]=timee;
17     for (auto i:graph[v]){
18         if(!visited[i])
19             dfs1(i);
20     }
21     vertex_color[color[v]-1].push_back(v);
22     timee++;
23     tout[v]=timee;
24 }
25 void dfs2(int v){
26     visited1[v]=true;
27     vertex_color2[color[v]-1].push_back(v);
28     for (auto i:graph[v]){
29         if(!visited1[i])
30             dfs2(i);
31     }
32 }
33
34 int main(){
35     int n,par;
36     cin>>n;
37     set<int> colors;
38     ;
39
40     for (int i=0;i<n-1;i++){
41         cin>>par;
42         graph[par-1].push_back(i+1);
43     }
44     for (int i=0;i<n;i++){
45         cin>>color[i];
46         colors.insert(color[i]-1);
47     }
48     dfs1(0);
49
50     dfs2(0);
51
52
53     for (auto x:colors){
54         if (vertex_color[x].size()>1){
55             if ((( tin[vertex_color[x][0]]<tin[vertex_color2[x].back()] )&&
56                  ( tout[vertex_color[x][0]]<tout[vertex_color2[x].back()] ) ) ||
57                  (( tin[vertex_color[x][0]]>tin[vertex_color2[x].back()] )&&
58                   ( tout[vertex_color[x][0]]>tout[vertex_color2[x].back()] ) ) ) {
59                 cout << "yes" << endl << vertex_color2[x].back()+1 << " " << vertex_color[x][0]+1;
60                 exit(0);
61             }
62         }
63     }
64     cout<<"no";
65 }
66
67
```

Task 8 - Result Documentation Report and Outcomes Placement Activities

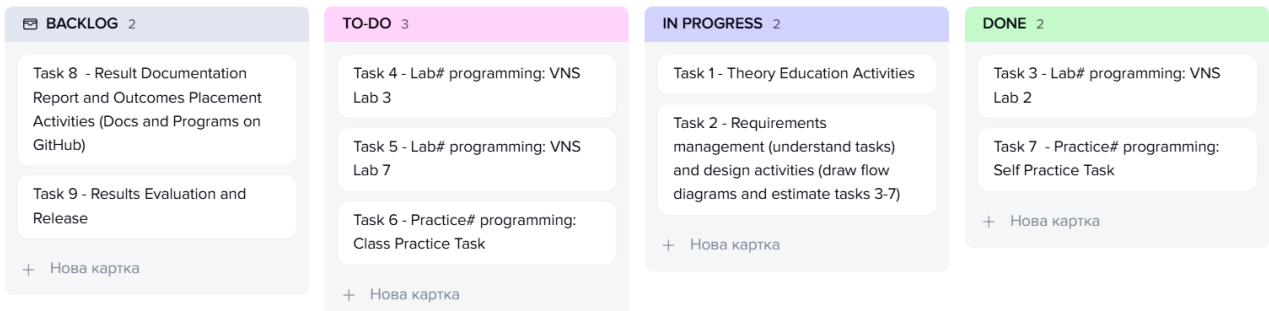
Час виконання ~ 1,5 год.



EPIC 3 Lesia EPIC 3 Dmytro EPIC 3 Yurii EPIC 1 Maksym

Enter your tasks and set priorities thanks to the properties: status, priority, urgency

Kanban board Board by Urgency Deadline



Pull Request

Висновок:

В результаті виконання цієї роботи я навчився ефективно використовувати цикли та вкладені цикли для реалізації повторюваних операцій у програмуванні. Опанував принципи роботи функцій, їх перевантаження, а також простір імен для покращення структури та організації коду. Навчився використовувати функції з змінною кількістю параметрів, що дозволило створювати більш гнучкі програми. Також засвоїв основи рекурсії для вирішення складних задач та навчився застосовувати вбудовані функції для полегшення стандартних операцій