



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

Перхун Максим Віталійович

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

Мета роботи:

Вдосконалити вміння програмувати шляхом вивчення циклів і функцій. Засвоїти різні види циклів (for, while, do-while) та навчитися керувати їх виконанням за допомогою операторів break і continue.

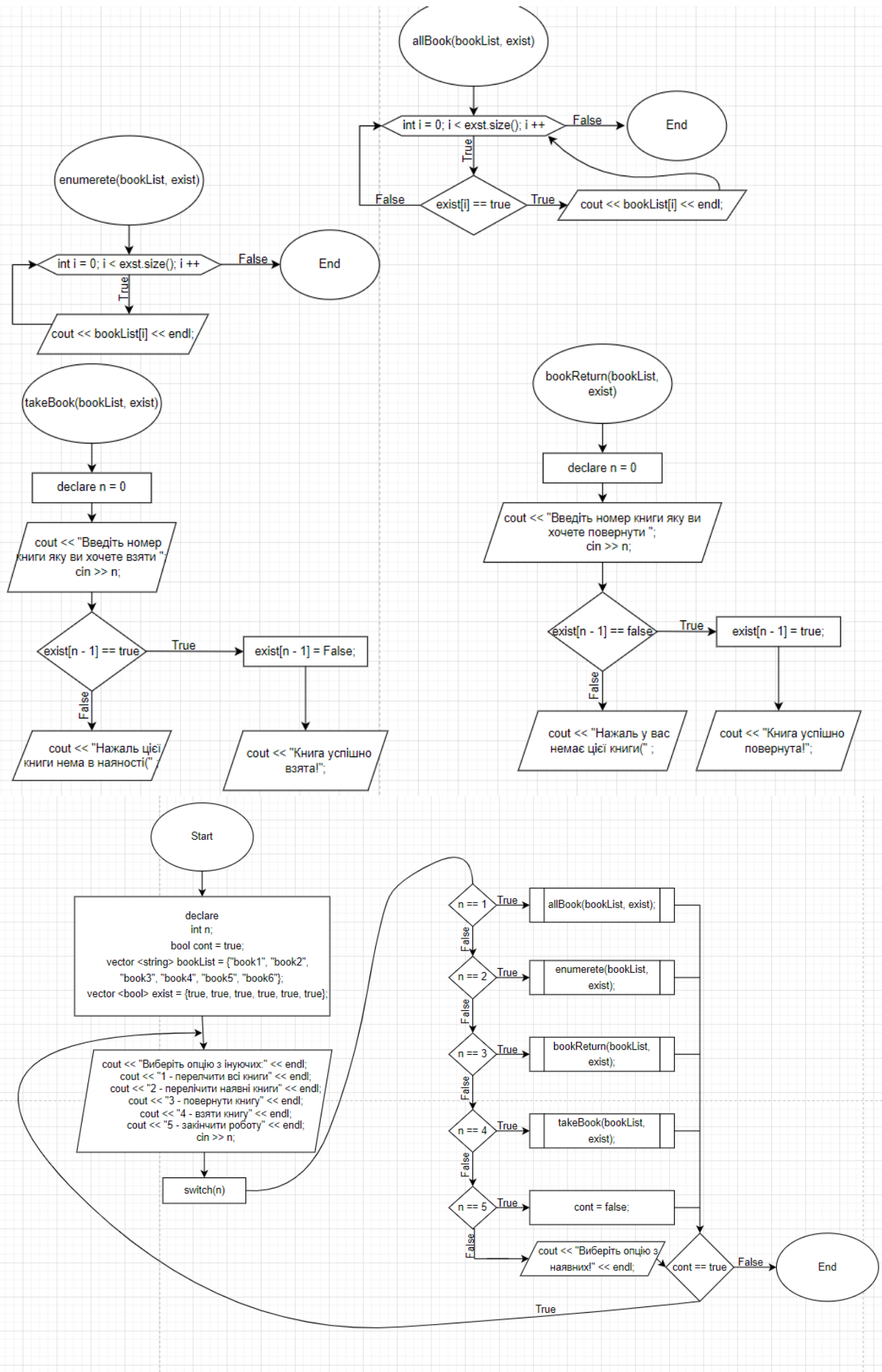
Навчитися працювати з функціями: оголошення, передача параметрів, перевантаження та рекурсія. Розібратися з еліпсисом та просторами імен для ефективної організації коду. Ознайомитися з використанням вбудованих функцій і функцій зі змінною кількістю параметрів.

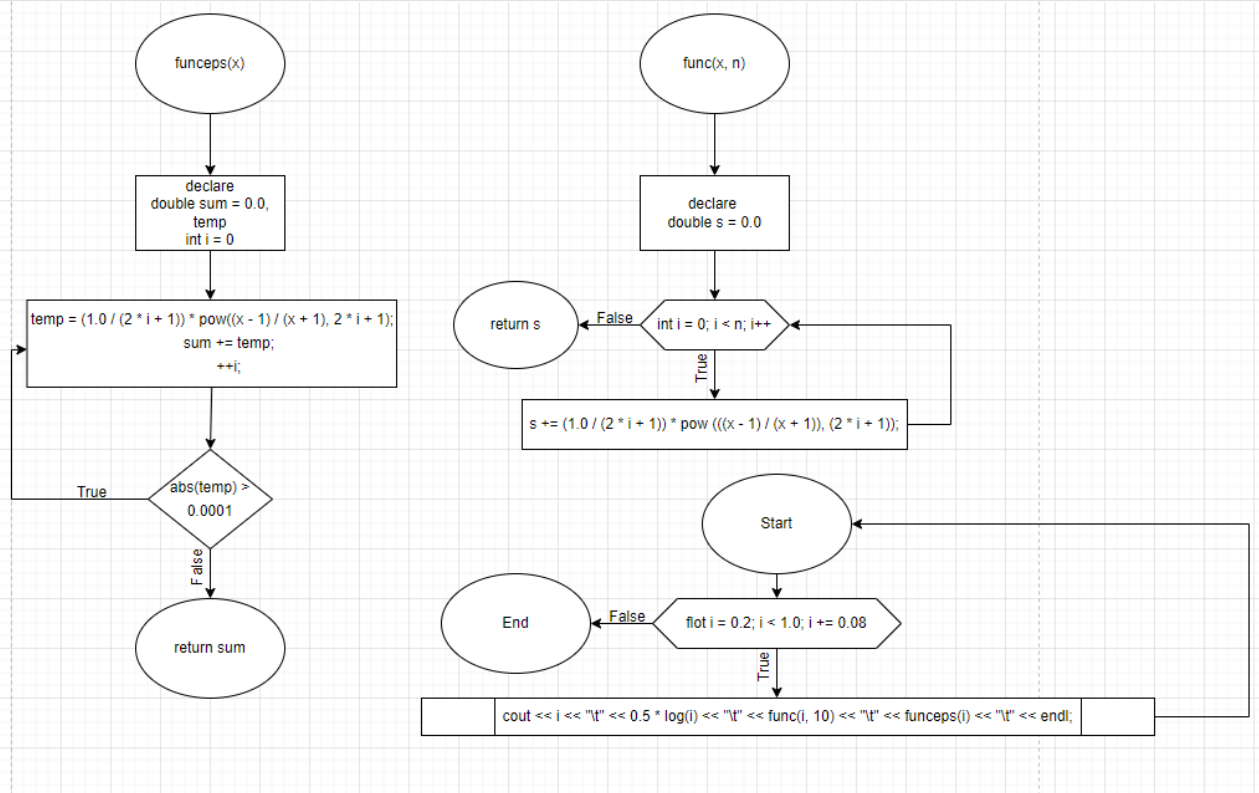
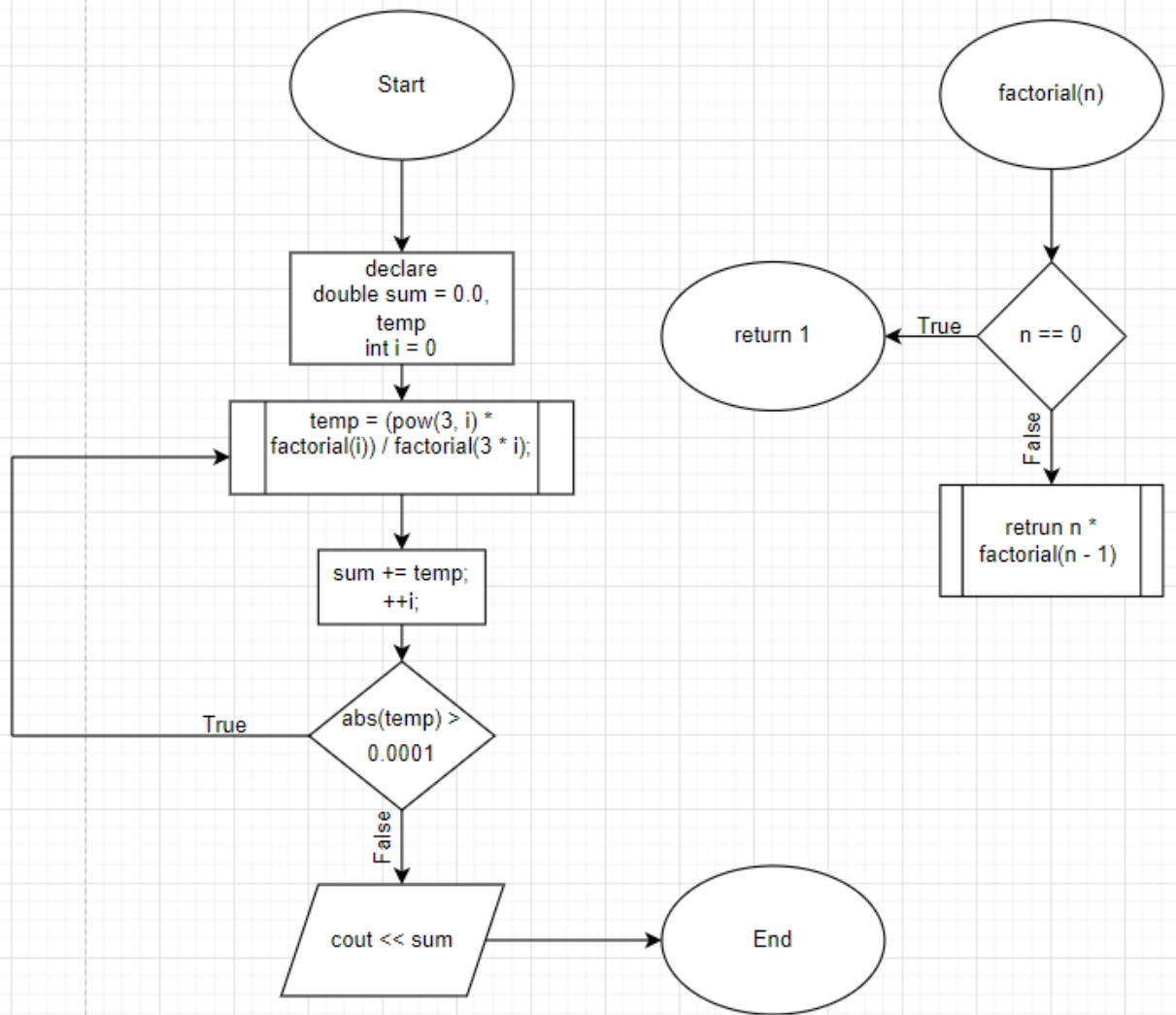
Теоретичні відомості:

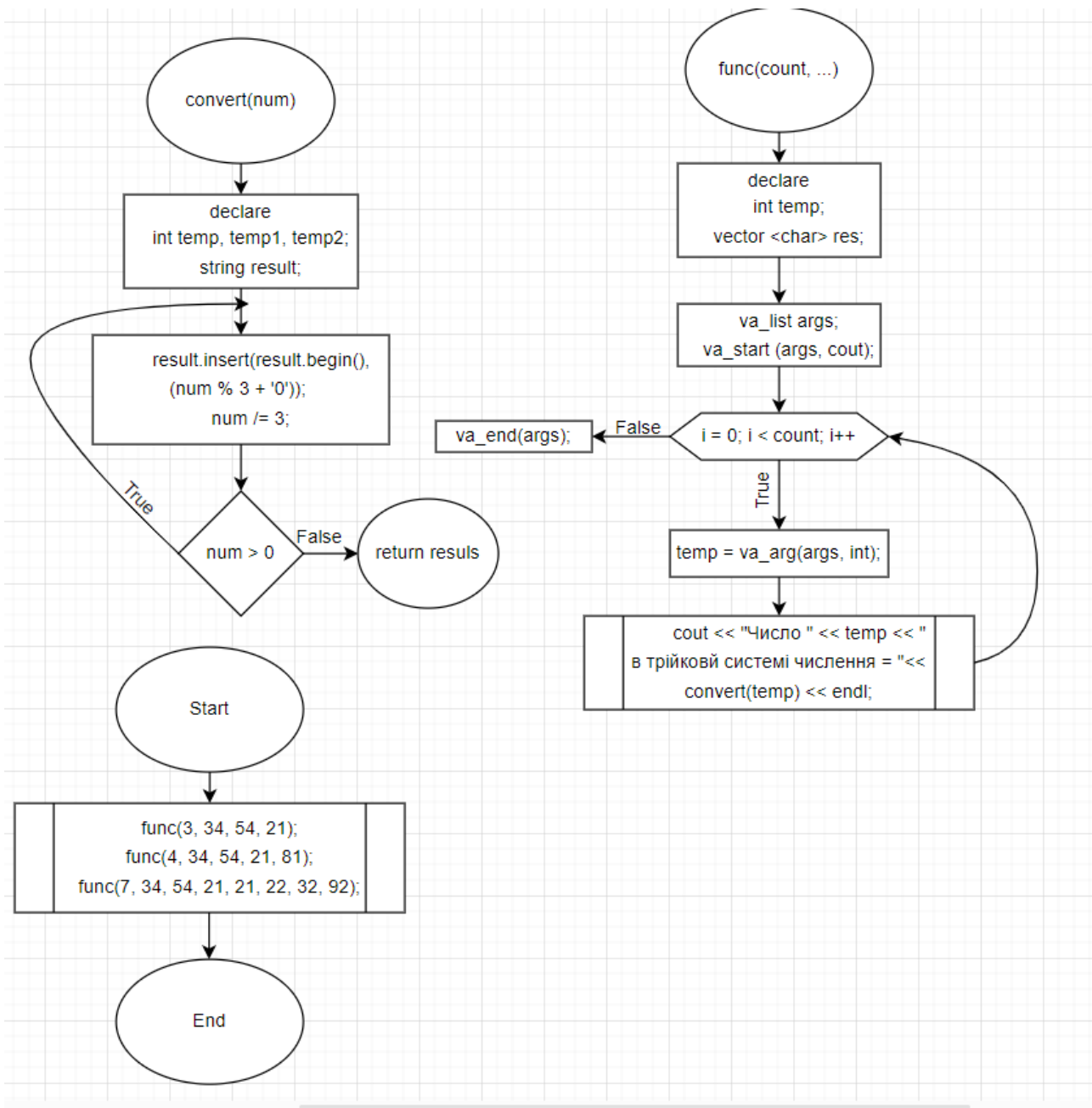
1. Теоретичні відомості з переліком важливих тем:
 - Тема №1: Введення в Цикли та їх Види в C++.
[C++ • Теорія • Урок 31 • Поняття циклу](#)
 - Тема №2: Управління Виконанням Циклів.
[C++ Break and Continue](#)
 - Тема №3: Вкладені Цикли.
[C++ • Теорія • Урок 38 • Вкладені конструкції](#)
 - Тема №4: Основи Функцій у C++.
[C++ Functions](#)
 - Тема №5: Перевантаження Функцій та Простір Імен.
[C++ • Теорія • Урок 48 • Перезавантаження функцій](#)
 - Тема №6: Розширені Можливості Функцій.
[C++ • Теорія • Урок 44 • Область видимості, глобальні/локальні дані, static/extern](#)
 - Тема №7: Вбудовані Функції в C++.
[Inline Functions in C++](#)

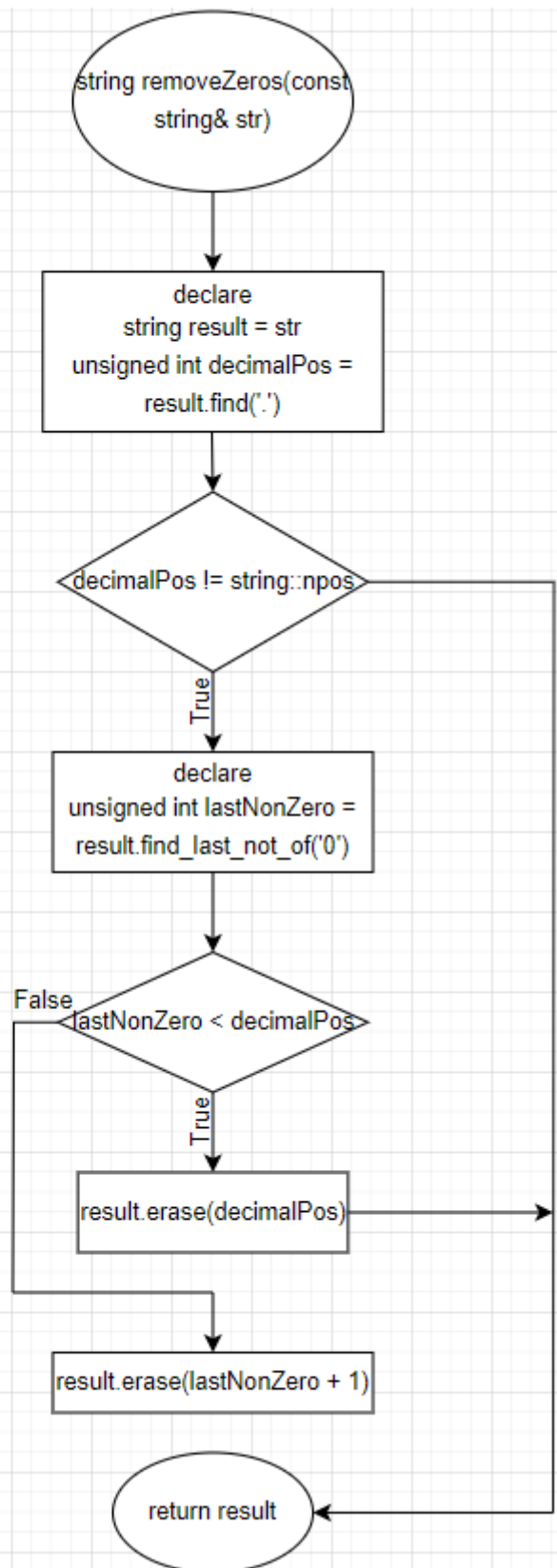
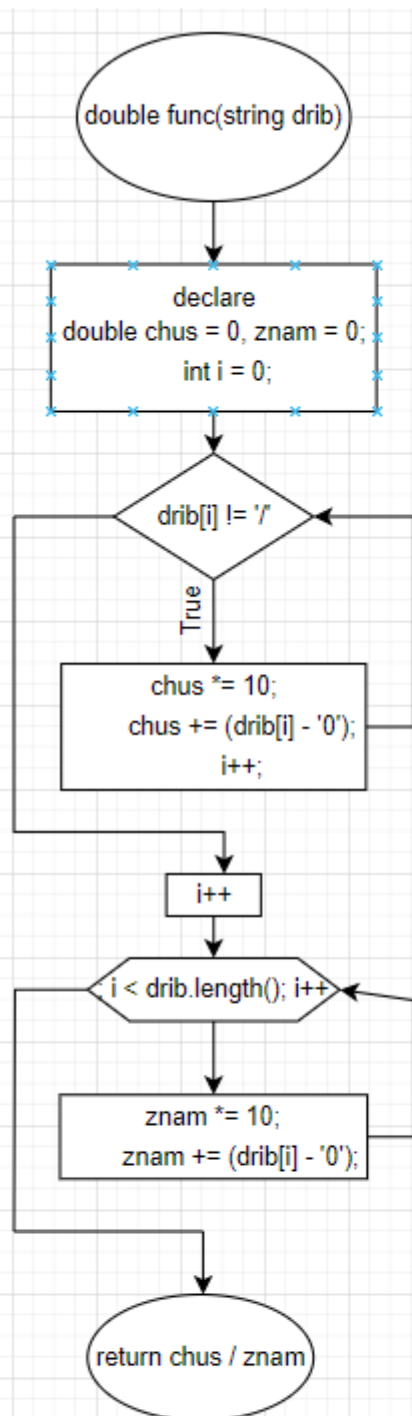
Виконання роботи

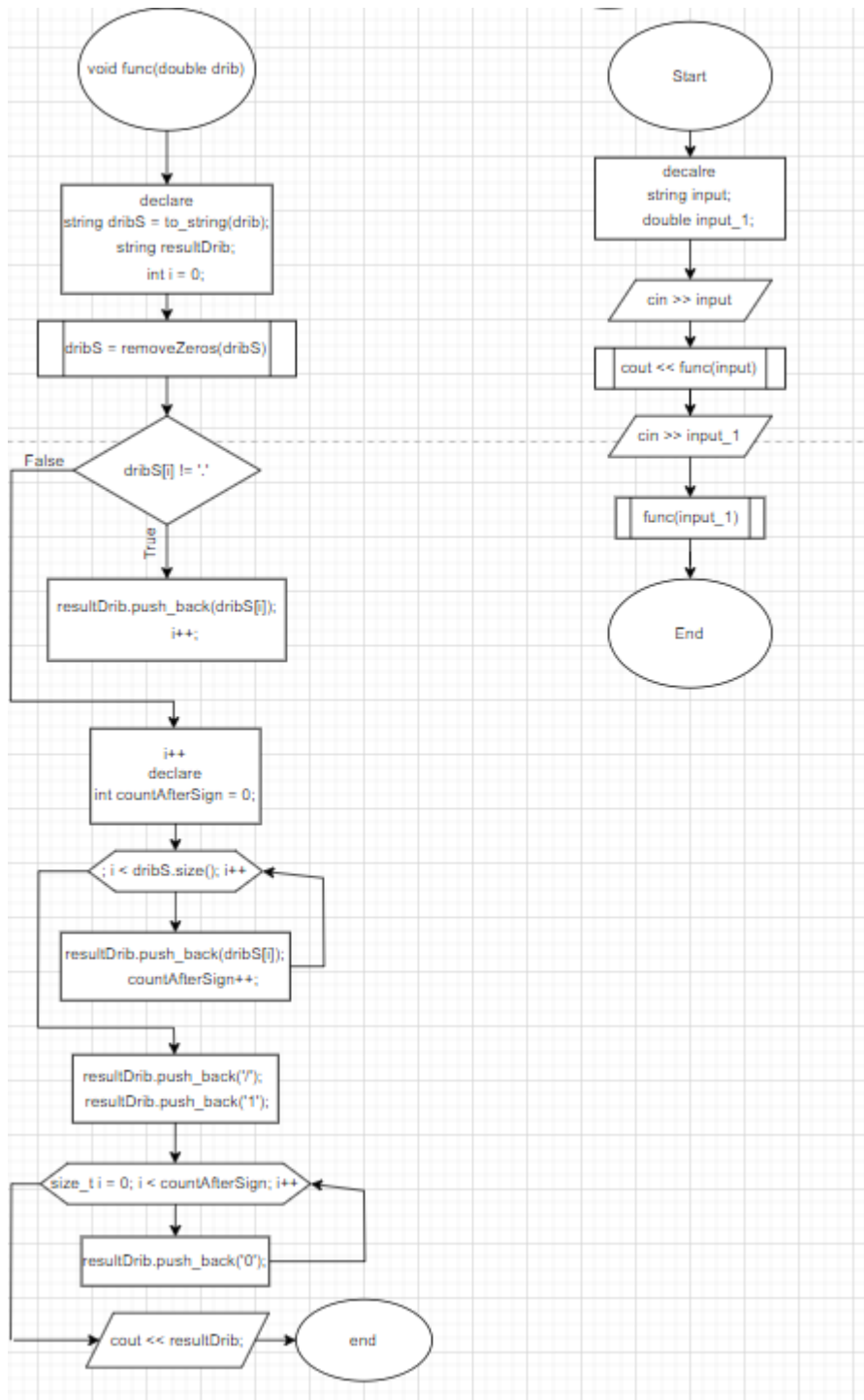
- Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)(4 год)

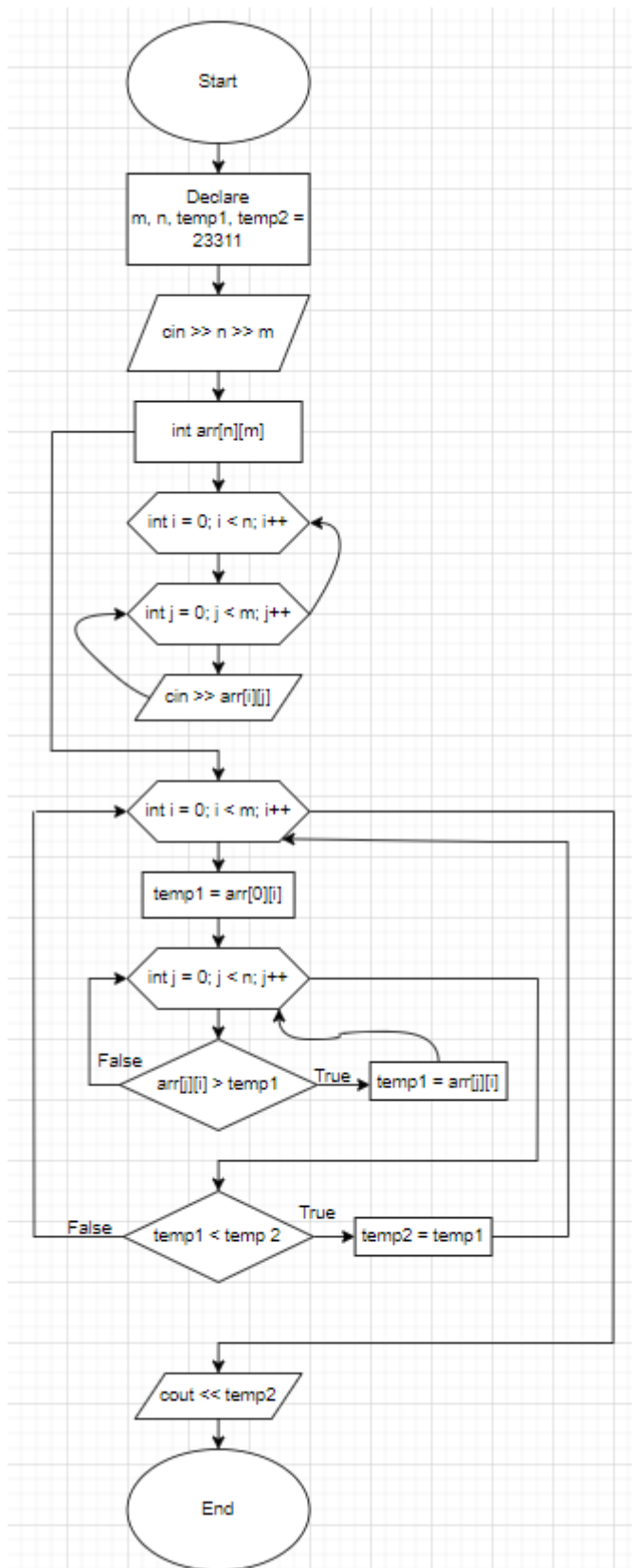












- Task 3 - Lab# programming: VNS Lab 2(20 xB)


```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int factorial(int n){
5      if (n == 0){
6          return 1;
7      }
8      return n * factorial(n-1);
9  }
10 int main() {
11     double sum = 0.0;
12     double temp;
13     int i = 0;
14     do {
15         temp = (pow(3, i) * factorial(i)) / factorial(3 * i);
16         sum += temp;
17         ++i;
18     } while (abs(temp) > 0.0001);
19     cout << sum;
20     return 0;
21 }
22

```

- Task 4 - Lab# programming: VNS Lab 3(40 xB)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  double funceps (float x){
5      double sum = 0.0;
6      double temp;
7      int i = 0;
8      do {
9          temp = (1.0 / (2 * i + 1)) * pow((x - 1) / (x + 1), 2 * i + 1);
10         sum += temp;
11         ++i;
12     } while (abs(temp) > 0.0001);
13     return sum;
14 }
15 double func (float x, int n){
16     double s = 0.0;
17     for (int i = 0; i < n; i++){
18         s += (1.0 / (2 * i + 1)) * pow (((x - 1) / (x + 1)), (2 * i + 1));
19     }
20     return s;
21 }
22
23
24 int main() {
25     for (float i = 0.2; i <= 1; i += 0.08){
26         cout << i << "\t" << 0.5 * log(i) << "\t" << func(i, 10) << "\t" << funceps(i) << "\t" << endl;
27     }
28     return 0;
29 }
30

```

- Task 5 - Lab# programming: VNS Lab 7(3 год)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  string convert(int num){
4      int temp, temp1, temp2;
5      string result;
6      do{
7          result.insert(result.begin(), (num % 3 + '0'));
8          num /= 3;
9      }while(num > 0);
10     return result;
11 }
12 void func(int count, ...){
13     int temp;
14     vector <char> res;
15     va_list args;
16     va_start (args, cout);
17     for (int i = 0; i < count; i++){
18         temp = va_arg(args, int);
19         cout << "Число " << temp << " в трійковій системі числення = "<< convert(temp) << endl;
20     }
21     va_end(args);
22 }
23
24 int main(){
25     func(3, 34, 54, 21);
26     func(4, 34, 54, 21, 81);
27     func(7, 34, 54, 21, 21, 22, 32, 92);
28     return 0;
29 }

```

Task 2

```

1  #include <iostream>
2  #include <vector>
3  #include <format>
4  #include <string>
5  #include<sstream>
6
7  using namespace std;
8  double func(string drib) {
9      double chus = 0, znam = 0;
10     int i = 0;
11     while (drib[i] != '/') {
12         chus *= 10;
13         chus += (drib[i] - '0');
14         i++;
15     }
16     i++;
17     for (; i < drib.length(); i++) {
18         znam *= 10;
19         znam += (drib[i] - '0');
20     }
21     return chus / znam;
22 }
23
24 string removeZeros(const string& str) {
25     string result = str;
26
27     unsigned int decimalPos = result.find('.');
28     if (decimalPos != string::npos) {
29         unsigned int lastNonZero = result.find_last_not_of('0');
30
31         if (lastNonZero < decimalPos) {
32             result.erase(decimalPos);
33         }
34         else {
35             result.erase(lastNonZero + 1);
36         }
37     }
38     return result;
39 }
40 }

```

```

42 void func(double drib)
43 {
44     string dribS = to_string(drib);
45     dribS = removeZeros(dribS);
46     string resultDrib;
47     int i = 0;
48     while (dribS[i] != '.')
49     {
50         resultDrib.push_back(dribS[i]);
51         i++;
52     }
53     i++;
54     int countAfterSign = 0;
55     for (; i < dribS.size(); i++)
56     {
57         resultDrib.push_back(dribS[i]);
58         countAfterSign++;
59     }
60     resultDrib.push_back('/');
61     resultDrib.push_back('1');
62     for (size_t i = 0; i < countAfterSign; i++)
63     {
64         resultDrib.push_back('0');
65     }
66
67     cout << resultDrib;
68 }
69
70 int main() {
71     string input;
72     cin >> input;
73     cout << func(input);
74     double input_1;
75     cin >> input_1;
76     func(input_1);
77     return 0;
78 }

```

- Task 6 - Practice# programming: Class Practice Task(2 год)

```

1  #include <iostream>
2  #include <vector>
3  #include <string>
4  using namespace std;
5
6  void enumerate(vector<string> &bookList, vector<bool> &exist){
7      for (int i = 0; i < exist.size(); i++){
8          if (exist[i] == true){
9              cout << bookList[i] << endl;
10         }
11     }
12 }
13 void allBook(vector<string> &bookList, vector<bool> &exist){
14     for(int i = 0; i < bookList.size(); i++){
15         cout << bookList[i] << endl;
16     }
17 }
18 void takeBook(vector<string> &bookList, vector<bool> &exist){
19     int n;
20     cout << "Введіть номер книги яку ви хочете взяти ";
21     cin >> n;
22     if (exist[n - 1] == true){
23         exist[n - 1] = false;
24         cout << "Книга успішно взята!" << endl;
25     }
26     else{
27         cout << "Нажаль цієї книги нема в наявності(" << endl;
28     }
29 }
30 void bookReturn(vector<string> &bookList, vector<bool> &exist){
31     int n;
32     cout << "Введіть номер книги яку ви хочете повернути ";
33     cin >> n;
34     if (exist[n - 1] == false){
35         exist[n - 1] = true;
36         cout << "Книга успішно повернута!" << endl;
37     }
38     else{
39         cout << "Нажаль ви вас не має цієї книги(" << endl;
40     }
41 }

```

```

42 int main(){
43     int n;
44     bool cont = true;
45     vector<string> bookList = {"book1", "book2", "book3", "book4", "book5", "book6"};
46     vector<bool> exist = {true, true, true, true, true, true};
47     do{
48         menu:
49         cout << "Виберіть опцію з інуючих:" << endl;
50         cout << "1 - перелчити всі книги" << endl;
51         cout << "2 - перелічити наявні книги" << endl;
52         cout << "3 - повернути книгу" << endl;
53         cout << "4 - взяти книгу" << endl;
54         cout << "5 - закінчити роботу" << endl;
55         cin >> n;
56         switch (n){
57             case 1:
58                 allBook(bookList, exist);
59                 break;
60             case 2:
61                 enumerate(bookList, exist);
62                 break;
63             case 3:
64                 bookReturn(bookList, exist);
65                 break;
66             case 4:
67                 takeBook( bookList, exist);
68                 break;
69             case 5:
70                 cont = false;
71                 break;
72             default:
73                 cout << "Виберіть опцію з наявних!" << endl;
74                 goto menu;
75                 break;
76         }
77     }while(cont);
78     return 0;
79 }
80

```

- Task 7 - Practice# programming: Self Practice Task(30 хв)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int main(){
4      int n, m, temp1, temp2 = 23131;
5      cin >> n >> m;
6      int arr[n][m];
7      for (int i = 0; i < n; i++){
8          for (int j = 0; j < m; j++){
9              cin >> arr[i][j];
10         }
11     }
12     for (int i = 0; i < m; i++){
13         temp1 = arr[0][i];
14         for (int j = 0; j < n; j++){
15             if(arr[j][i] > temp1){
16                 temp1 = arr[j][i];
17             }
18         }
19         if (temp2 > temp1){
20             temp2 = temp1;
21         }
22     }
23     cout << temp2;
24     return 0;
25 }

```

Висновки:

По завершенню завдань Епіка №3, я вдосконалити вміння програмувати шляхом вивчення циклів і функцій. Засвоїла різні види циклів (for, while, do-while) та навчилася керувати їх виконанням за допомогою операторів break і continue. Навчилася працювати з функціями: оголошення, передача параметрів, перевантаження та рекурсія. Розібралася з еліпсисом та просторами імен для ефективної організації коду. Ознайомилася з використанням вбудованих функцій і функцій зі змінною кількістю параметрів.