

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 № 2,3,7

Практичних Робіт до блоку № 3

Виконав:

Студент групи ШІ-12

Токарник Сергій

Львів 2024

Тема роботи

1. Використання циклів, вкладених циклів та завершення їх виконання.
2. Використання функцій та перенавантажених функцій, а також функцій з змінною кількістю параметрів, використання вбудованих функцій.

Мета роботи

1. Навчитись використовувати вкладені та звичайні цикли та завершувати їх виконання.
2. Навчитись користуватись звичайними та перенавантаженими функціями, функціями з змінною кількістю параметрів, правильно їх викликати.

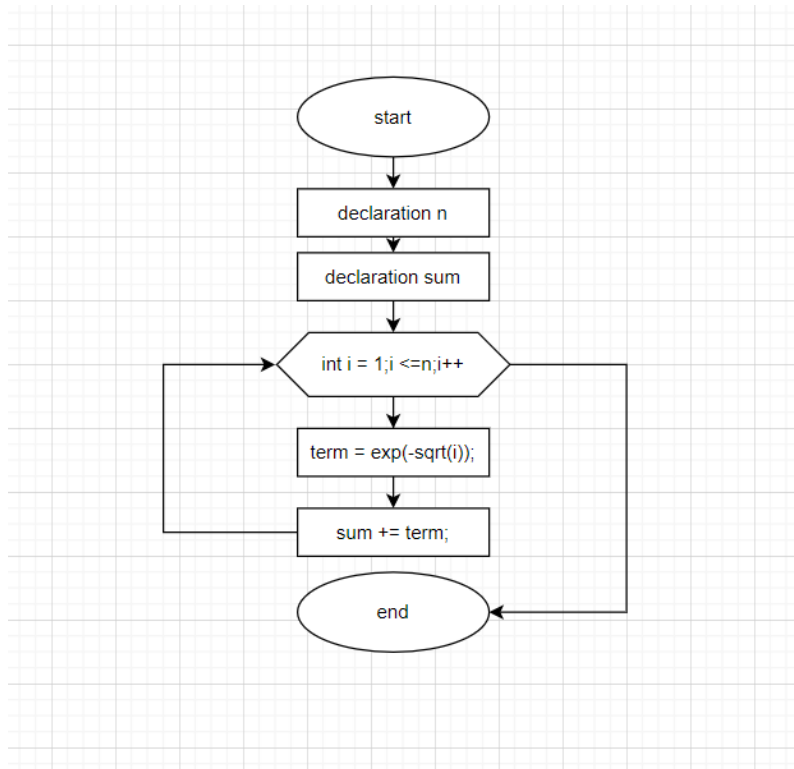
Теоретичні відомості

1. <https://www.quora.com/How-do-you-round-a-floating-point-number-to-one-digit-after-the-decimal-place-in-C>
2. <https://stackoverflow.com/questions/50453557/c-how-to-delete-a-specific-row-or-column-in-a-dynamically-allocated-2d-array>
3. <https://acode.com.ua/urok-70-tsykl-while/>
4. <https://acode.com.ua/urok-71-tsykl-do-while/>
5. <https://acode.com.ua/urok-69-operator-goto/>
6. <https://acode.com.ua/urok-102-parametry-i-argumenty-funktsij/>
7. <https://acode.com.ua/urok-108-perevantazhennya-funktsij/>
8. <https://acode.com.ua/urok-107-vbudovani-funktsiyi/>

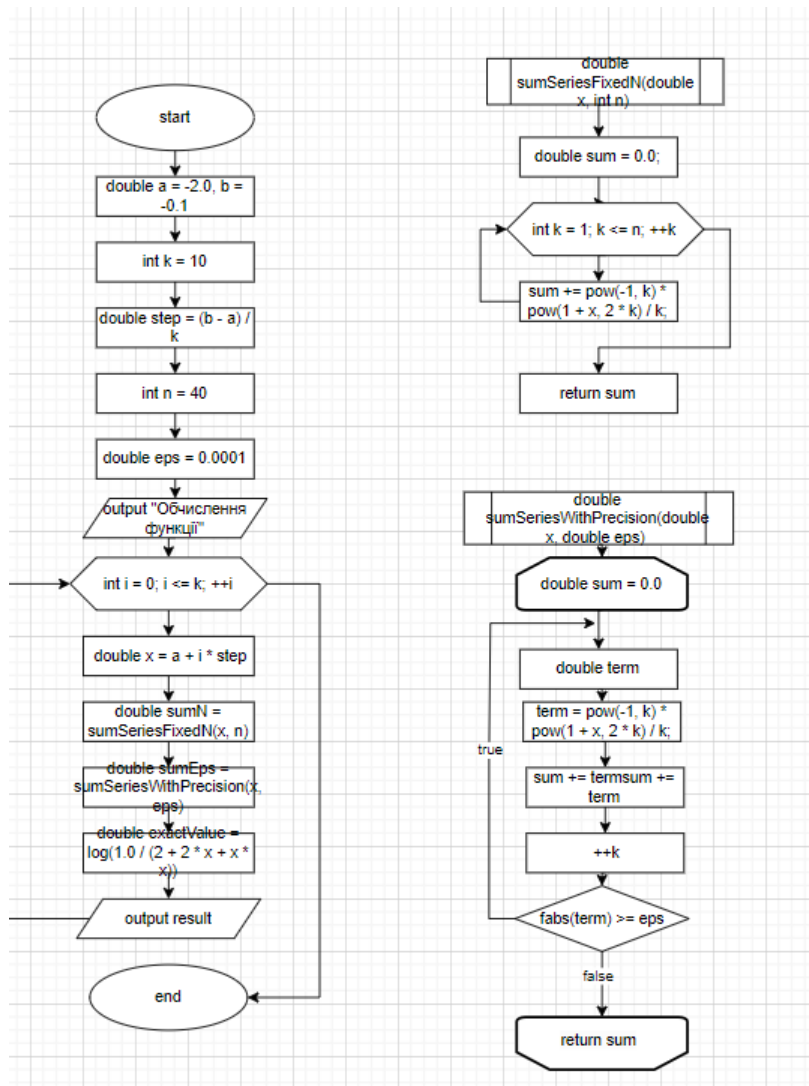
Виконання роботи

Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7) (5год)

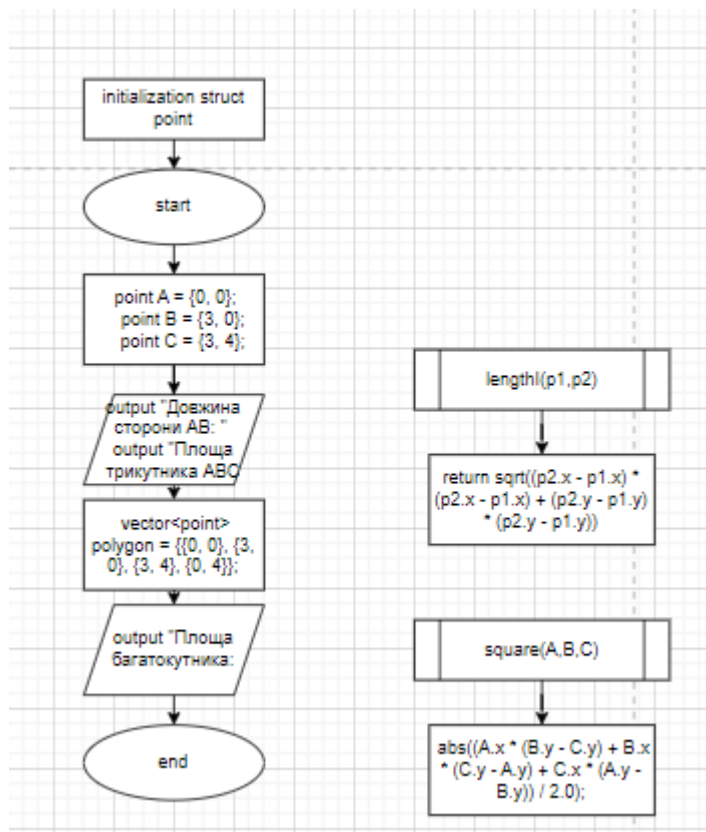
1) VNS Lab 2

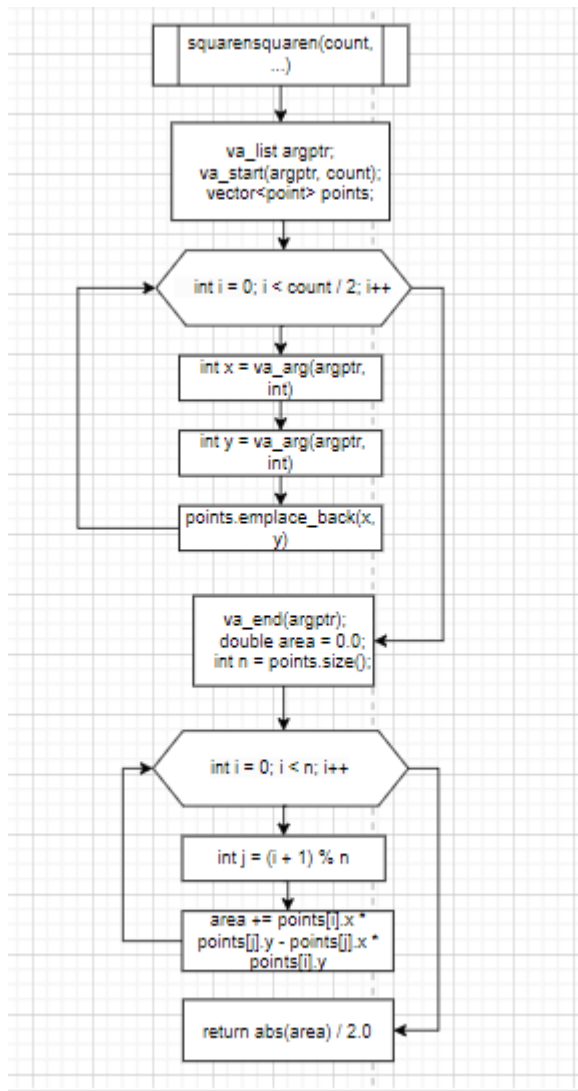


2) VNS Lab 3

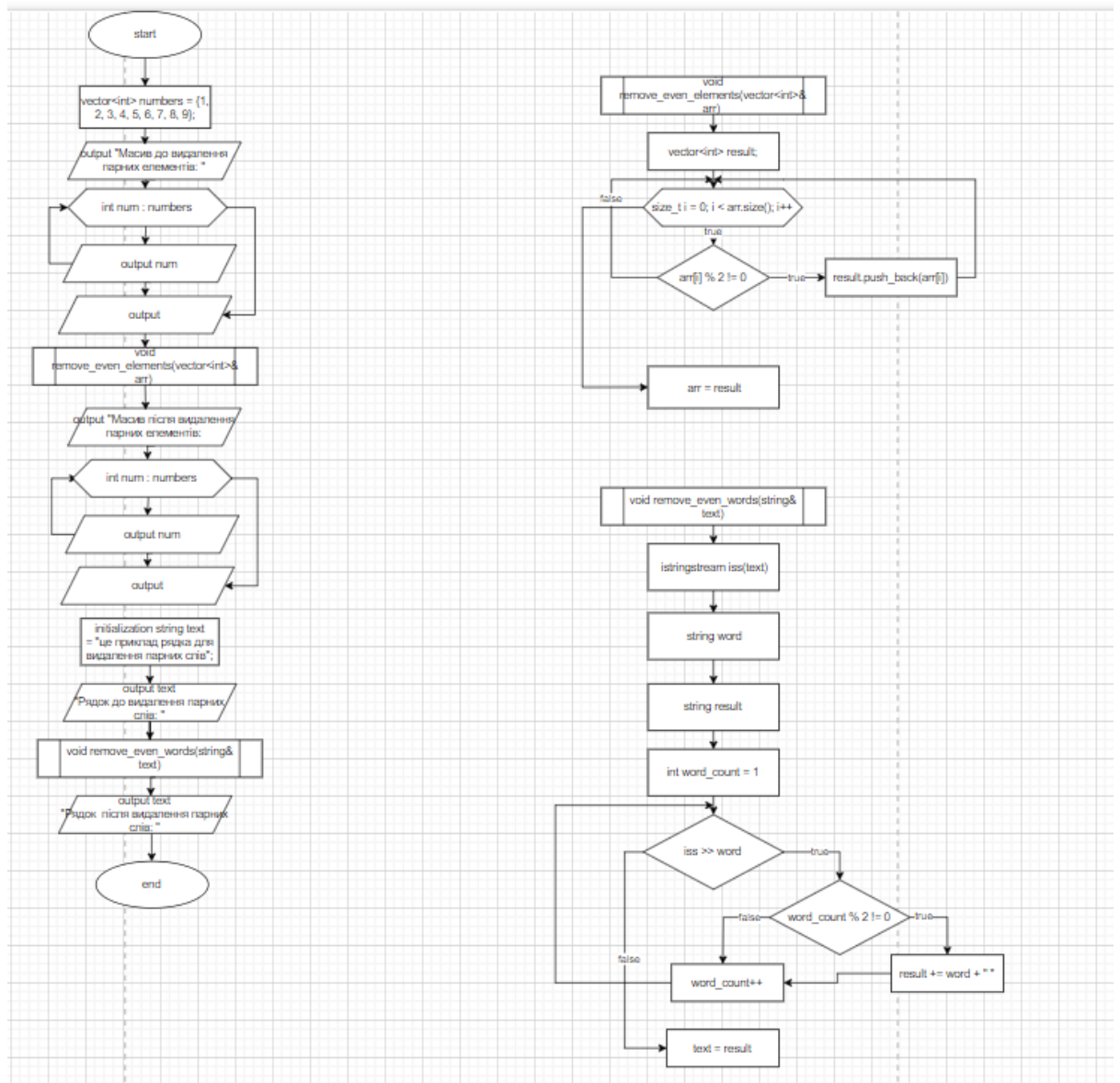


3) VNS Lab 7 Task 1

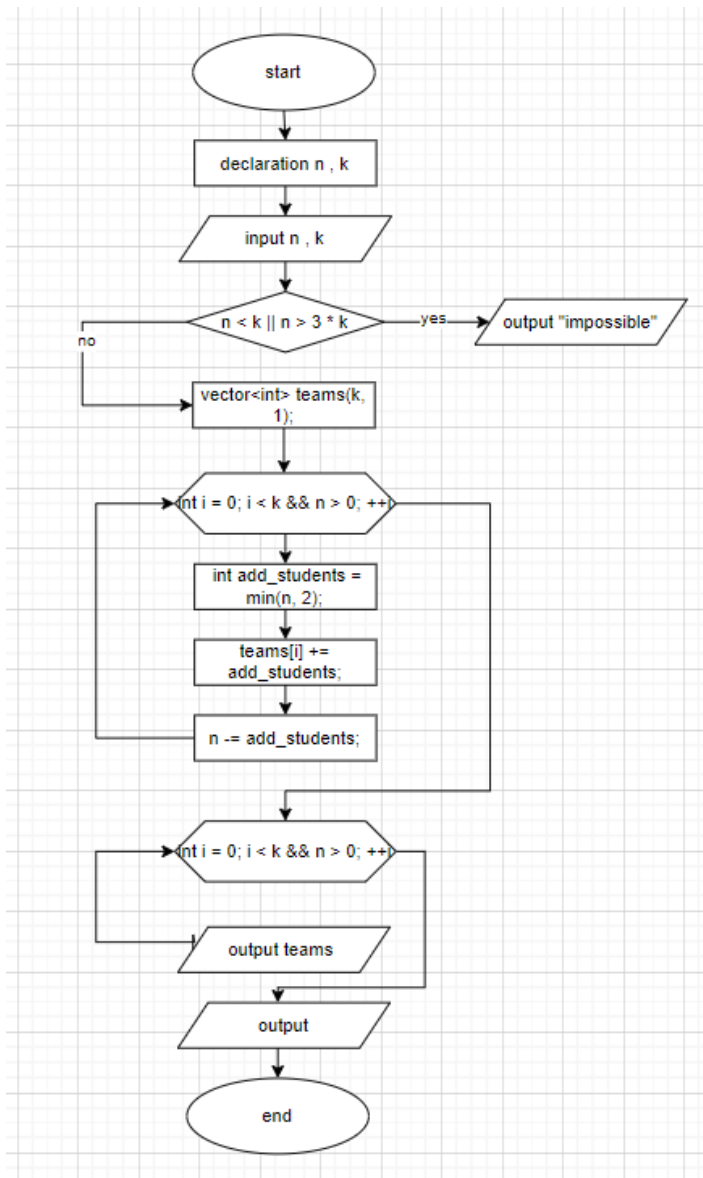




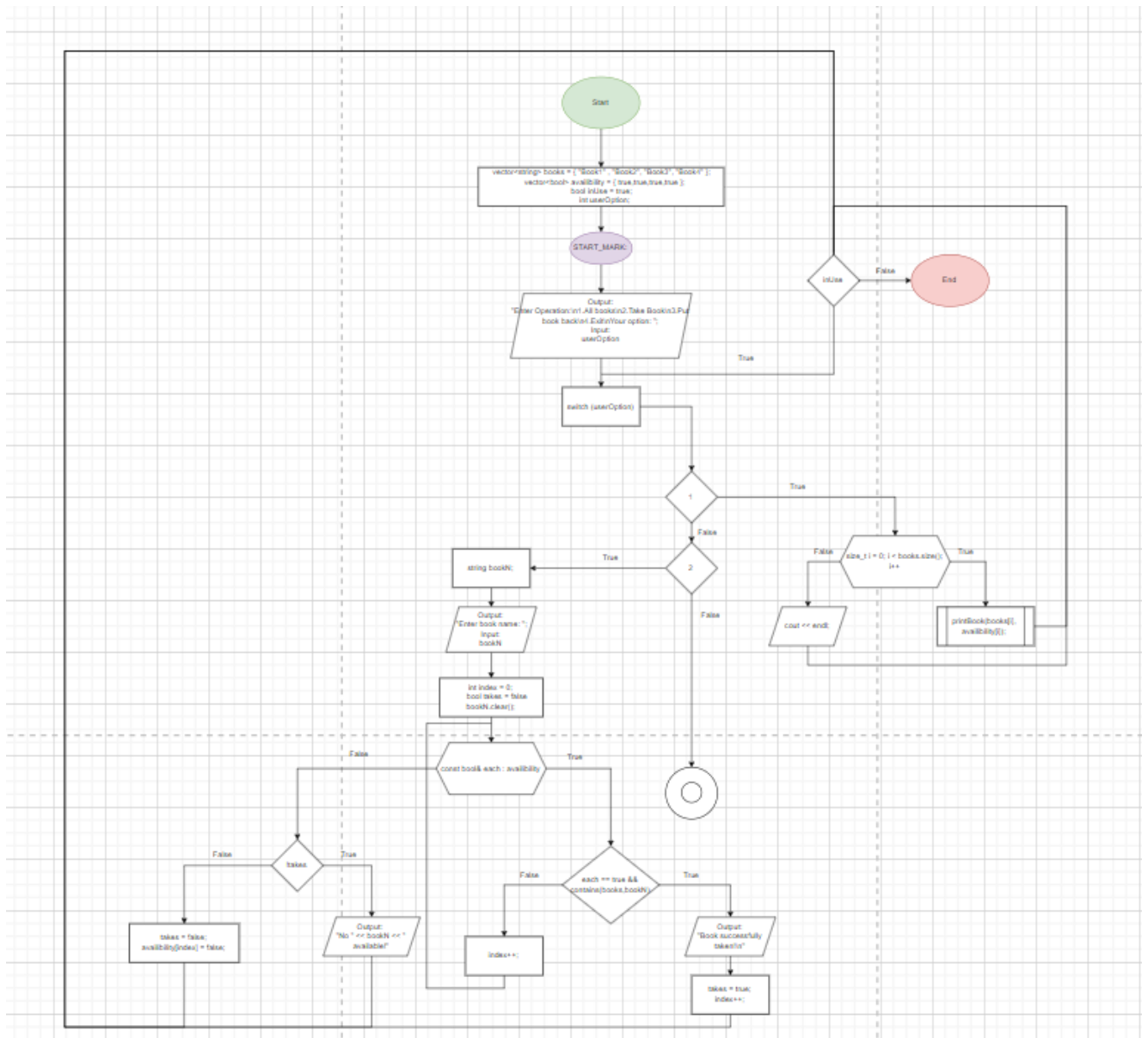
4) VNS Lab 7 Task 2

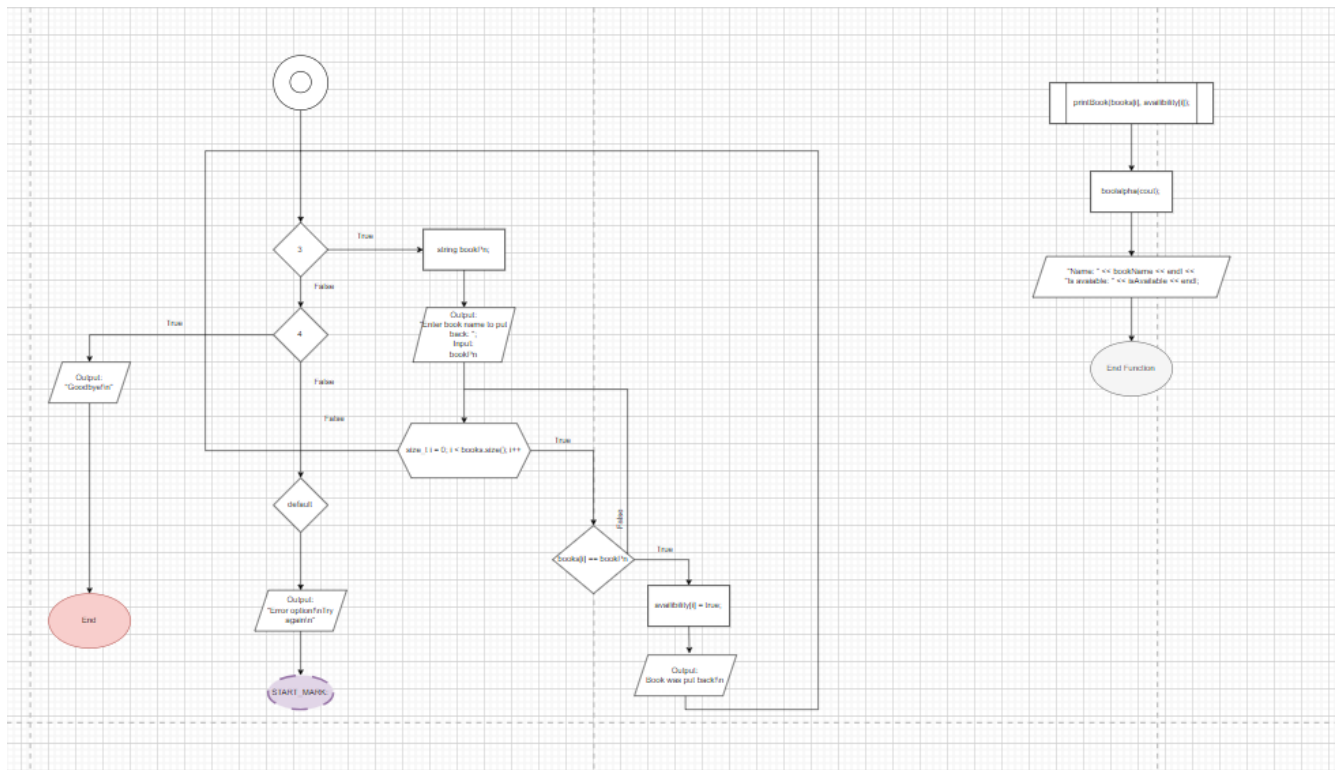


5) Self Practice Work



6)Class Practice Task





(all codes 3год30хв)

Task 3 – Lab# programming: VNS Lab 2(15хв)

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7
8      int n = 9;
9      double sum = 0.0;
10
11     for (int i = 1; i <= n; i++) {
12         double term = exp(-sqrt(i));
13         sum += term;
14     }
15
16     cout << "The sum of the first 9 terms is: " << sum << endl;
17
18     return 0;
19
20 }
  
```

Task 4 - Lab# programming: VNS Lab 3(30xв)

```
1  #include <iostream>
2  #include <cmath>
3  #include <iomanip>
4  #include <vector>
5
6  using namespace std;
7
8  double sumSeriesFixedN(double x, int n) {
9      double sum = 0.0;
10     for (int k = 1; k <= n; ++k) {
11         sum += pow(-1, k) * pow(1 + x, 2 * k) / k;
12     }
13     return sum;
14 }
15
16 double sumSeriesWithPrecision(double x, double eps) {
17     double sum = 0.0;
18     double term;
19     int k = 1;
20     do {
21         term = pow(-1, k) * pow(1 + x, 2 * k) / k;
22         sum += term;
23         ++k;
24     } while (fabs(term) >= eps);
25     return sum;
26 }
27
28 int main() {
29     double a = -2.0, b = -0.1;
30     int k = 10;
31     double step = (b - a) / k;
32     int n = 40;
33     double eps = 0.0001;
34
35     cout << "Обчислення функції" << endl;
36     cout << "-----" << endl;
37
38     for (int i = 0; i <= k; ++i) {
39         double x = a + i * step;
40
41         double sumN = sumSeriesFixedN(x, n);
42
43         double sumEps = sumSeriesWithPrecision(x, eps);
44
45         double exactValue = log(1.0 / (2 + 2 * x + x * x));
46
47         cout << fixed << setprecision(5);
48         cout << "X=" << x << "\tSN=" << sumN << "\tSE=" << sumEps << "\tY=" << exactValue << endl;
49     }
50
51     return 0;
52 }
53
```

Task 5 - Lab# programming: VNS Lab 7(task 1 , task 2)(1год)

Task1

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <cmath>
4  #include <vector>
5  #include <stdarg.h>
6
7  using namespace std;
8
9
10 struct point{
11     int x, y;
12     point(int x, int y) : x(x), y(y) {}
13 };
14
15 int length(point& p1, point& p2){
16     return sqrt((p2.x - p1.x) * (p2.x - p1.x) + (p2.y - p1.y) * (p2.y - p1.y));
17 }
18
19 int square(point& A, point& B, point& C) {
20     return abs(A.x * (B.y - C.y) + B.x * (C.y - A.y) + C.x * (A.y - B.y)) / 2.0;
21 }
22
23 double squaren(int count, ...)
24 {
25     va_list argptr;
26     va_start(argptr, count);
27
28     vector<point> points;
29     for (int i = 0; i < count / 2; i++) {
30         int x = va_arg(argptr, int);
31         int y = va_arg(argptr, int);
32         points.emplace_back(x, y);
33     }
34
35     va_end(argptr);
36
37     double area = 0.0;
38     int n = points.size();
39     for (int i = 0; i < n; i++) {
40         int j = (i + 1) % n;
41         area += points[i].x * points[j].y - points[j].x * points[i].y;
42     }
43
44     return abs(area) / 2.0;
45 }
46
47 int main() {
48     point A = {0, 0};
49     point B = {3, 0};
50     point C = {3, 4};
51
52     cout << "Длина стороны AB: " << length(A, B) << endl;
53     cout << "Площа трикутника ABC: " << square(A, B, C) << endl;
54
55     vector<point> polygon = {{0, 0}, {3, 0}, {3, 4}, {0, 4}};
56     cout << "Площа багатокутника: " << squaren(8, 0, 0, 3, 0, 3, 4, 0, 4) << endl;
57
58     return 0;
59 }
```

Task2

```
1  #include <iostream>
2  #include <vector>
3  #include <sstream>
4
5  using namespace std;
6
7  void remove_even_elements(vector<int>& arr) {
8      vector<int> result;
9      for (size_t i = 0; i < arr.size(); i++) {
10         if (arr[i] % 2 != 0) {
11             result.push_back(arr[i]);
12         }
13     }
14     arr = result;
15 }
16
17 void remove_even_words(string& text) {
18     istringstream iss(text);
19     string word;
20     string result;
21     int word_count = 1;
22     while (iss >> word) {
23         if (word_count % 2 != 0) {
24             result += word + " ";
25         }
26         word_count++;
27     }
28     text = result;
29 }
30
31 int main() {
32
33     vector<int> numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9};
34     cout << "Масив до видалення парних елементів: ";
35     for (int num : numbers) cout << num << " ";
36     cout << endl;
37
38     remove_even_elements(numbers);
39
40     cout << "Масив після видалення парних елементів: ";
41     for (int num : numbers) cout << num << " ";
42     cout << endl;
43
44     string text = "це приклад рядка для видалення парних слів";
45     cout << "Рядок до видалення парних слів: " << text << endl;
46
47     remove_even_words(text);
48
49     cout << "Рядок після видалення парних слів: " << text << endl;
50
51     return 0;
52 }
```

Task 6 - Practice# programming: Class Practice Task(1год30хв)

```

1  #include<iostream>
2  #include<vector>
3  #include<string>
4  using namespace std;
5
6
7  void printBook(string bookName, bool isAvailable)
8  {
9      boolalpha(cout);
10     cout << "Name: " << bookName << endl << "Is available: " << isAvailable << endl;
11 }
12
13 bool contains(vector<string> books , string book)
14 {
15     for (auto & b : books)
16     {
17         if(b == book)
18             return true;
19     }
20     return false;
21 }
22
23 bool isAvailable(const vector<string>& books , string book, vector<bool>& availList)
24 {
25     for (size_t i = 0; i < books.size(); i++)
26     {
27         if(books[i] == book)
28         {
29             return availList[i];
30         }
31     }
32     return false;
33 }
34
35 int main()
36 {
37     vector<string> books = { "Book1" , "Book2", "Book3", "Book4" };
38     vector<bool> availability = { true,true,true,true };
39
40     bool inUse = true;
41     int userOption;
42     do
43     {
44         START_MARK:
45         cout << "Enter Operation:\n1.All books\n2.Take Book\n3.Put book back\n4.Exit\nYour option: ";
46         cin >> userOption;
47         switch (userOption)
48         {
49             case 1:
50             {
51                 for (size_t i = 0; i < books.size(); i++)
52                 {
53                     printBook(books[i], availability[i]);
54                 }
55                 cout << endl;

```

```

55         cout << endl;
56         break;
57     }
58     case 2:
59     {
60         string bookN;
61         cout << "Enter book name: ";
62         cin >> bookN;
63         int index = 0;
64         bool takes = false;
65         bookN.clear();
66         for(const bool& each : availability)
67         {
68             if(each == true && contains(books,bookN))
69             {
70                 cout << "Book successfully taken!\n";
71                 takes = true;
72                 index++;
73                 break;
74             }
75             index++;
76         }
77         if(!takes)
78         {
79             cout << "No " << bookN << " available!";
80             break;
81         }
82         takes = false;
83         availability[index] = false;
84         break;
85     }
86     case 3:
87     {
88         string bookPn;
89         cout << "Enter book name to put back: ";
90         cin >> bookPn;
91         for (size_t i = 0; i < books.size(); i++)
92         {
93             if(books[i] == bookPn)
94             {
95                 availability[i] = true;
96                 cout << "Book was put back!\n";
97                 break;
98             }
99         }
100         break;
101     }
102     case 4:
103     {
104         cout << "Goodbye!\n";
105         cout << "Goodbye!\n";
106         // inUse = false;
107         // goto END;
108         return 0;
109         break;
110     }
111     default:
112     {
113         cout << "Error option!\nTry again\n";
114         goto START_MARK;
115         break;
116     }
117 } while(inUse);
118
119 END:
120 return 0;
121 }

```

Task 7 - Practice# programming: Self Practice Task(15xb)

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      int n, k;
7      cin >> n >> k;
8
9      if (n < k || n > 3 * k) {
10         cout << "Impossible" << endl;
11         return 0;
12     }
13
14     vector<int> teams(k, 1);
15     n -= k;
16
17     for (int i = 0; i < k && n > 0; ++i) {
18         int add_students = min(n, 2);
19         teams[i] += add_students;
20         n -= add_students;
21     }
22
23     for (int i = 0; i < k; ++i) {
24         cout << teams[i] << " ";
25     }
26     cout << endl;
27
28     return 0;
29 }

```

Зустрічі з командою



Висновок: під час виконання епіку №3 я навчився краще робити та використовувати функції та цикли.