

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Кафедра систем штучного інтелекту



## **Звіт**

**про виконання розрахунково-графічних робіт блоку № 7**

**з дисципліни:** «Основи програмування»

до:

ВНС Розрахунково-графічних робіт № 1-4

Практичних Робіт до блоку № 7

**Виконав(ла):**

Студент(ка) групи ШІ-12

Олійник Божена Орестівна

Львів 2024

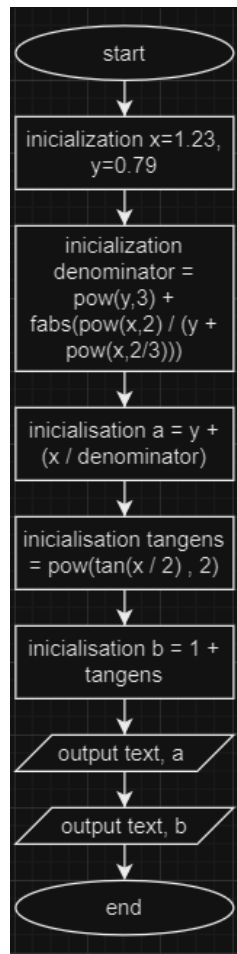
**Тема: перевірка знань здобутих за цей семестр, а саме: функцій вводу та виводу, циклів (з умовою, з передумовою та з після умовою), умовних операторів, одновимірних та двовимірних масивів, створення та ініціалізація змінних та констант, створення та виклик функції, перенавантаження функції, рекурсивна функція, використання структур, виконання унарних та бінарних операцій.**

**Мета: перевірити себе на знання з вищенаведених тем.**

### VNS practice work task №1

```
1  #include <stdio.h>
2  #include <iostream>
3  #include <cmath>
4
5  using namespace std;
6
7  int main()
8  {
9      float x = 1.23, y = 0.79;
10
11     double denominator = pow(y, 3) + fabs(pow(x, 2) / (y + pow(x, 2 / 3)));
12
13     double a = y + (x / denominator);
14
15     double tangens = pow(tan(x / 2), 2);
16
17     double b = 1 + tangens;
18
19     cout << "a = " << a << endl;
20
21     cout << "b = " << b << endl;
22 }
```

```
a = 1.70912
b = 1.49898
```



## VNS practice work task №2

```

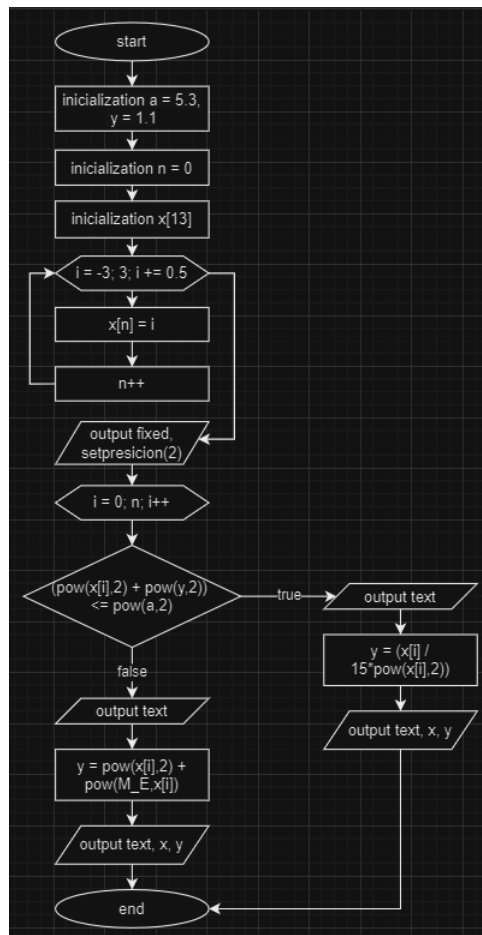
1  #include <stdio.h>
2  #include <iostream>
3  #include <iomanip>
4  #include <cmath>
5
6  using namespace std;
7
8  int main()
9  {
10     double a = 5.3, y = 1.1;
11     int n = 0;
12     double x[13];
13     for (double i = -3; i <= 3; i += 0.5){
14         x[n] = i;
15         n++;
16     }
17
18     cout << fixed << setprecision(2);
19
20     for (int i = 0; i < n; i++){
21         if ((pow(x[i],2) + pow(y,2)) <= pow(a,2)){
22             cout << "First case:" << endl;
23             y = (x[i]/(15*pow(x[i],2)));
24             cout << "x = " << x[i] << " " << '\t' << "y = " << y << endl;
25         } else {
26             cout << "Second case:" << endl;
27             y = pow(x[i],2) + pow(M_E,x[i]);
28             cout << "x = " << x[i] << " " << '\t' << "y = " << y << endl;
29         }
30     }
31 }
32

```

```

First case:
x = -3.00      y = -0.02
First case:
x = -2.50      y = -0.03
First case:
x = -2.00      y = -0.03
First case:
x = -1.50      y = -0.04
First case:
x = -1.00      y = -0.07
First case:
x = -0.50      y = -0.13
First case:
x = 0.00       y = nan
Second case:
x = 0.50       y = 1.90
First case:
x = 1.00       y = 0.07
First case:
x = 1.50       y = 0.04
First case:
x = 2.00       y = 0.03
First case:
x = 2.50       y = 0.03
First case:
x = 3.00       y = 0.02

```



## VNS practice work task №3

```

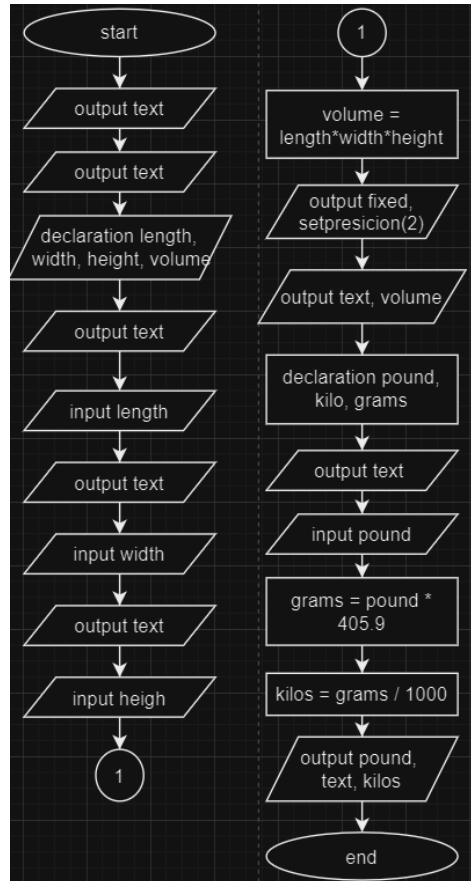
1  #include <iostream>
2  #include <iomanip>
3
4  using namespace std;
5
6  int main() {
7      cout << "Обчислення об'єму паралелепіпеда." << endl;
8      cout << "Введіть початкові дані:" << endl;
9
10     double length, width, height, volume;
11     cout << "Довжина (см) = ";
12     cin >> length;
13     cout << "Ширина(см) = ";
14     cin >> width;
15     cout << "Висота(см) = ";
16     cin >> height;
17
18     volume = length * width * height;
19
20     cout << fixed << setprecision(2);
21     cout << "Об'єм: " << volume << " куб.см." << endl;
22
23     double pound, kilos, grams;
24     cout << "Введіть вагу в фунтах: ";
25     cin >> pound;
26
27     grams = pound * 405.9;
28     kilos = grams / 1000;
29
30     cout << pound << " фунтів дорівнюють " << kilos << " кілограмам" << endl;
31
32     return 0;
33 }
34

```

```

Обчислення об'єму паралелепіпеда.
Введіть початкові дані:
Довжина (см) = 9
Ширина(см) = 7.5
Висота(см) = 5
Об'єм: 337.50 куб.см.
Введіть вагу у фунтах: 30
30.00 фунтів дорівнюють 12.18 кілограмам

```



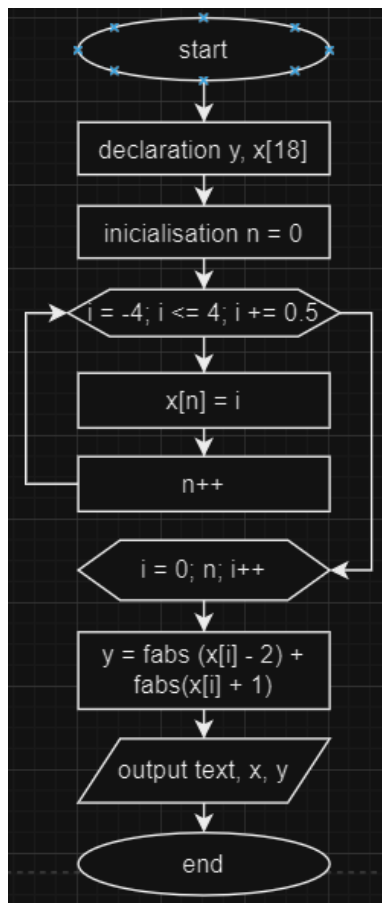
## VNS practice work task №4

```

1  #include <iostream>
2  #include <string>
3  #include <iomanip>
4  #include <cmath>
5  using namespace std;
6
7  int main()
8  {
9      double y, x[18];
10     int n = 0;
11     for (double i = -4; i <= 4; i += 0.5)
12     {
13         x[n] = i;
14         n++;
15     }
16
17     for (int i = 0; i < n; i++)
18     {
19         y = fabs(x[i] - 2) + fabs(x[i] + 1);
20         cout << "x = " << x[i] << " " << '\t' << "y = " << y << endl;
21     }
22 }

```

|          |       |
|----------|-------|
| x = -4   | y = 9 |
| x = -3.5 | y = 8 |
| x = -3   | y = 7 |
| x = -2.5 | y = 6 |
| x = -2   | y = 5 |
| x = -1.5 | y = 4 |
| x = -1   | y = 3 |
| x = -0.5 | y = 3 |
| x = 0    | y = 3 |
| x = 0.5  | y = 3 |
| x = 1    | y = 3 |
| x = 1.5  | y = 3 |
| x = 2    | y = 3 |
| x = 2.5  | y = 4 |
| x = 3    | y = 5 |
| x = 3.5  | y = 6 |
| x = 4    | y = 7 |



# Algotester problem №0522

0522 - Вболівальниці

C++ 23

Accepted

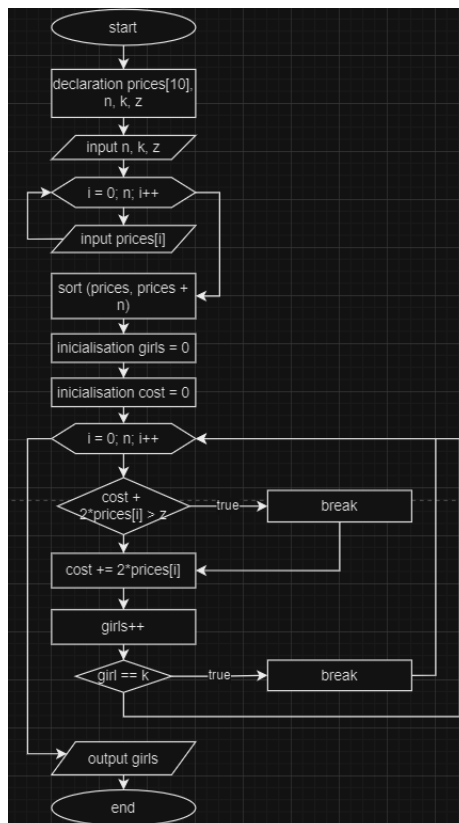
0.003

1.203

1863941

```
1  #include <iostream>
2  #include <algorithm>
3
4  using namespace std;
5
6  int main() {
7      int prices[10];
8      int n, k, z;
9
10     cin >> n >> k >> z;
11     for (int i = 0; i < n; i++) {
12         cin >> prices[i];
13     }
14
15     sort(prices, prices + n);
16     int girls = 0;
17     int cost = 0;
18
19     for (int i = 0; i < n; i++) {
20         if (cost + 2 * prices[i] > z) {
21             break;
22         }
23         cost += 2 * prices[i];
24         girls++;
25         if (girls == k) {
26             break;
27         }
28     }
29
30     cout << girls << endl;
31
32     return 0;
33 }
34
```

```
4 7 44
4 47 7 74
2
```





## Algotester problem №0023

0023 - Депутатські краватки

C++ 23

Accepted

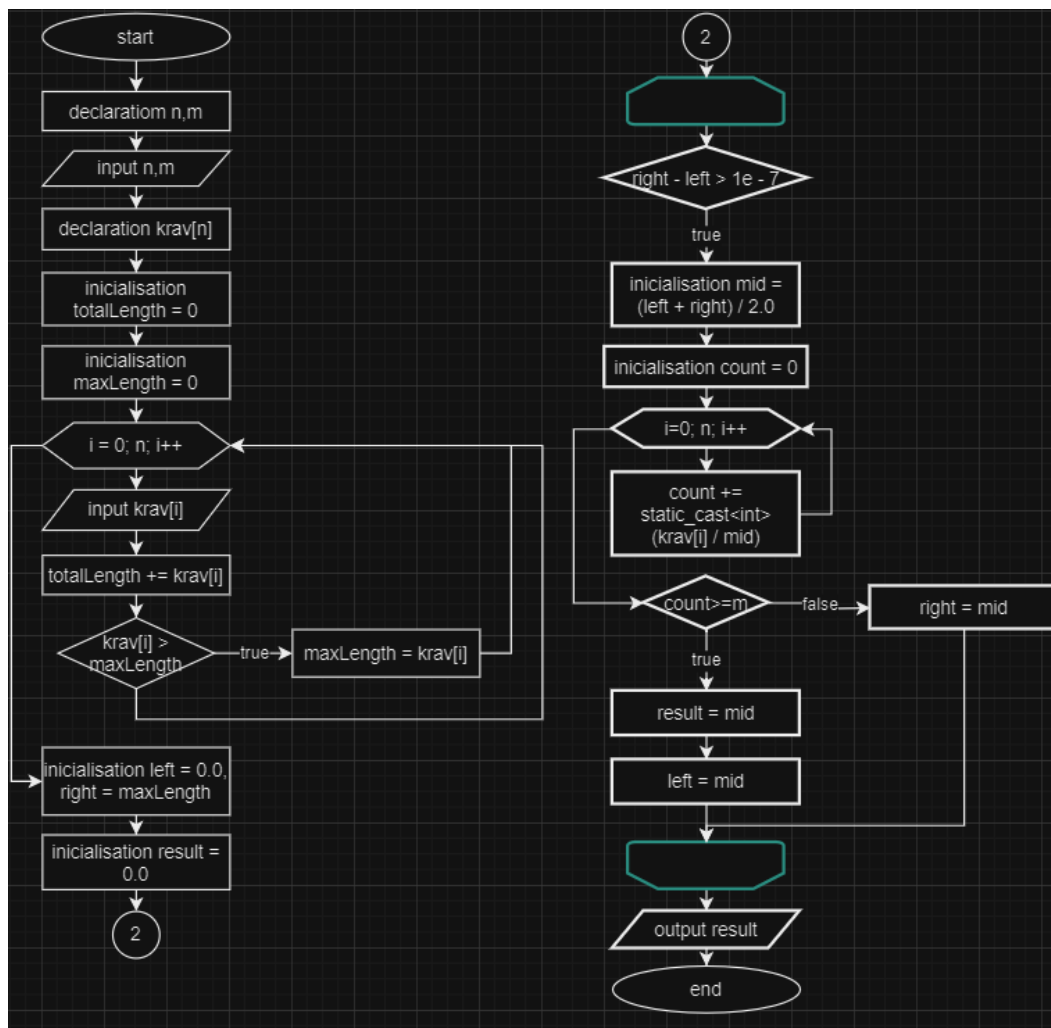
0.003

1.184

1864092

```
1  #include <iostream>
2  #include <iomanip>
3
4  using namespace std;
5
6  int main() {
7      int n, m;
8      cin >> n >> m;
9
10     int krav[n];
11     int totallength = 0;
12     int maxLength = 0;
13
14     for (int i = 0; i < n; i++) {
15         cin >> krav[i];
16         totallength += krav[i];
17         if (krav[i] > maxLength) {
18             maxLength = krav[i];
19         }
20     }
21
22     double left = 0.0, right = maxLength;
23     double result = 0.0;
24
25     while (right - left > 1e-7) {
26         double mid = (left + right) / 2.0;
27         int count = 0;
28
29         for (int i = 0; i < n; i++) {
30             count += static_cast<int>(krav[i] / mid);
31         }
32
33         if (count >= m) {
34             result = mid;
35             left = mid;
36         } else {
37             right = mid;
38         }
39     }
40
41     cout << fixed << setprecision(7) << result << endl;
42
43     return 0;
44 }
```

```
3 4
1 10 5
3.3333333
```



## Algotester problem №0183

0183 - Breath of Fire

C++ 23

Accepted

0.002

0.977 1864237

```

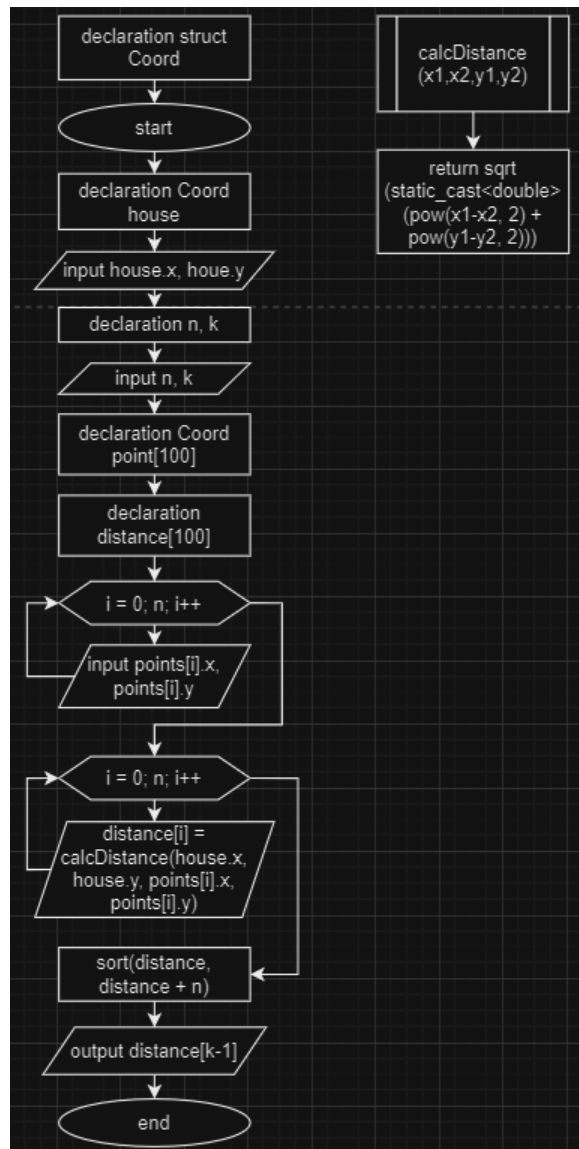
1  #include <iostream>
2  #include <cmath>
3  #include <algorithm>
4  #include <iomanip>
5  using namespace std;
6
7  struct Coord {
8      int x;
9      int y;
10 };
11
12 double calcDistance(int x1, int y1, int x2, int y2) {
13     return sqrt(static_cast<double>(pow(x1 - x2, 2) + pow(y1 - y2, 2)));
14 }
15
16 int main() {
17     Coord house;
18     cin >> house.x >> house.y;
19
20     int n, k;
21     cin >> n >> k;
22
23     Coord points[1000];
24     double distance[1000];
25
26     for (int i = 0; i < n; i++) {
27         cin >> points[i].x >> points[i].y;
28     }
29
30     for (int i = 0; i < n; i++) {
31         distance[i] = calcDistance(house.x, house.y, points[i].x, points[i].y);
32     }
33
34     sort(distance, distance + n);
35     cout << setprecision(13) << distance[k - 1];
36     return 0;
37 }

```

```

5 6
4 2
3 4
8 11
6 5
17 25
2.828427124746

```



## Algotester problem №0606

|             |        |          |       |       |         |
|-------------|--------|----------|-------|-------|---------|
| 0606 - Chir | C++ 23 | Accepted | 0.003 | 1.117 | 1864340 |
|-------------|--------|----------|-------|-------|---------|

```

1  #include <iostream>
2  #include <cmath>
3  #include <algorithm>
4  #include <cstring>
5
6  using namespace std;
7
8  const double INF = 1e9;
9
10 struct Edge {
11     int u, v;
12     double weight;
13 };
14
15 int n;
16 int parent[100];
17 Edge edges[5000];
18
19 int find(int x) {
20     if (parent[x] != x)
21         parent[x] = find(parent[x]);
22     return parent[x];
23 }
24
25 void unionSets(int x, int y) {
26     int rootX = find(x);
27     int rootY = find(y);
28     if (rootX != rootY)
29         parent[rootX] = rootY;
30 }
31
32 double distance(int i, int j, int x[], int y[]) {
33     return sqrt((x[i] - x[j]) * (x[i] - x[j]) + (y[i] - y[j]) * (y[i] - y[j]));
34 }
35
36 int main() {
37     cin >> n;
38     int x[n], y[n];
39
40     for (int i = 0; i < n; i++) {
41         cin >> x[i] >> y[i];
42         parent[i] = i;
43     }
44

```

```

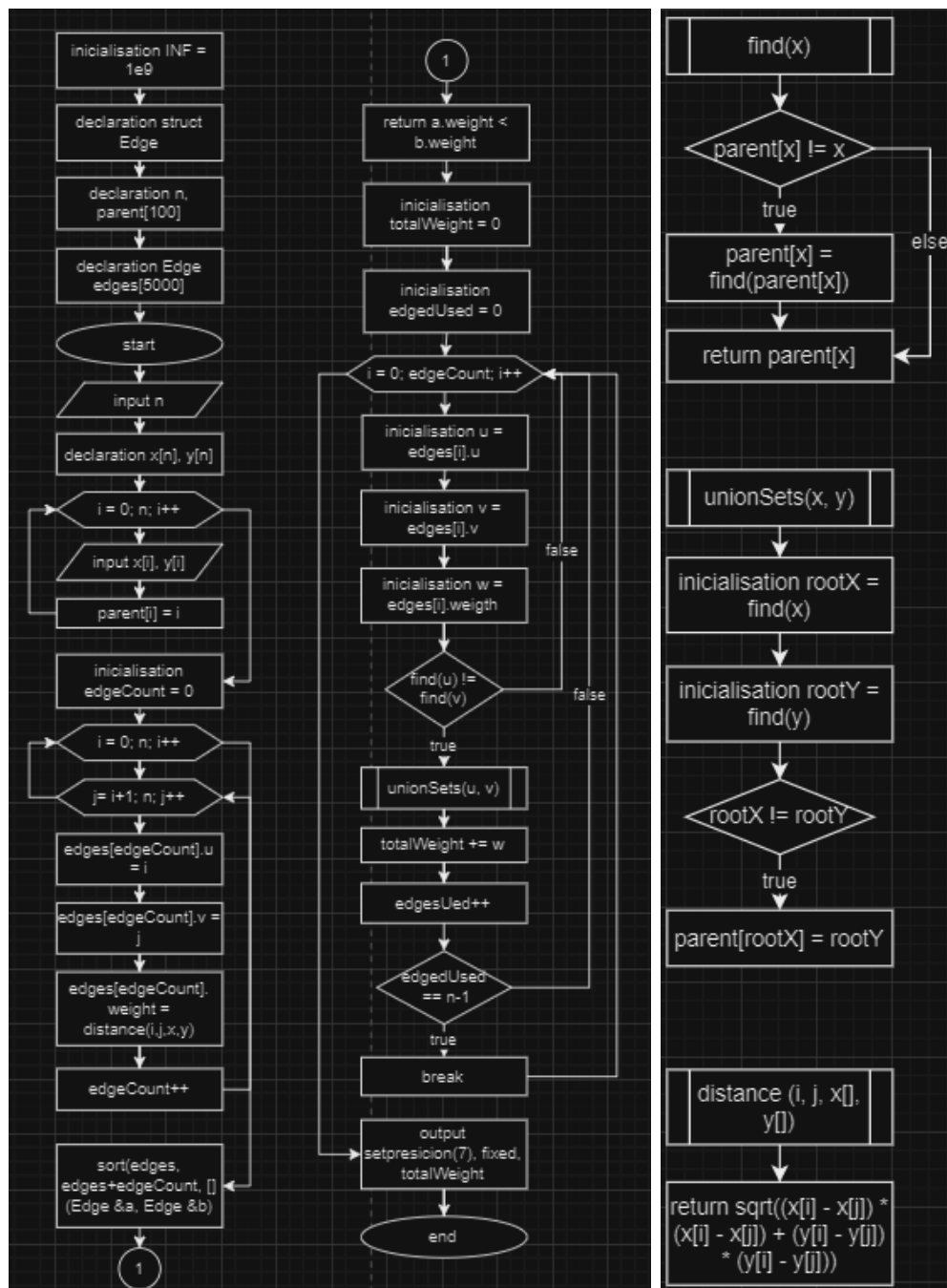
44
45     int edgeCount = 0;
46
47     for (int i = 0; i < n; i++) {
48         for (int j = i + 1; j < n; j++) {
49             edges[edgeCount].u = i;
50             edges[edgeCount].v = j;
51             edges[edgeCount].weight = distance(i, j, x, y);
52             edgeCount++;
53         }
54     }
55
56     sort(edges, edges + edgeCount, [](Edge &a, Edge &b) {
57         return a.weight < b.weight;
58     });
59
60     double totalWeight = 0;
61     int edgesUsed = 0;
62
63     for (int i = 0; i < edgeCount; i++) {
64         int u = edges[i].u;
65         int v = edges[i].v;
66         double w = edges[i].weight;
67
68         if (find(u) != find(v)) {
69             unionSets(u, v);
70             totalWeight += w;
71             edgesUsed++;
72             if (edgesUsed == n - 1) break;
73         }
74     }
75
76     cout.precision(7);
77     cout << fixed << totalWeight << endl;
78
79     return 0;
80 }

```

```

4
0 1
1 0
-1 0
0 -1
4.2426407

```



## Висновок

**В результаті виконання даної саги я переконалась, що вмію та знаю як використовувати цикли, умовні оператори, функції вводу та виводу, створювати та ініціалізувати змінні та константи, використовувати структури, створювати та викликати функції, виконувати унарні та бінарні операції, працювати з циклами.**