

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:
Студент групи ІІІ-12
Гаврих Юрій Дмитрович

Мета роботи:

Одержати практичні навички в розробці і дослідженні алгоритмів розв'язання задач.

Виконання роботи

Завдання №1

VNS Practice Work Task 1

Розробити лінійний алгоритм для розв'язання задачі.

Варіант 15. $\varphi = \sqrt{mtgx + |c \sin x|}$; $\xi = m \cos(bx + \sin x) + c^3$, де $m=2$; $c=1$; $x=1,2$; $b=0,7$.

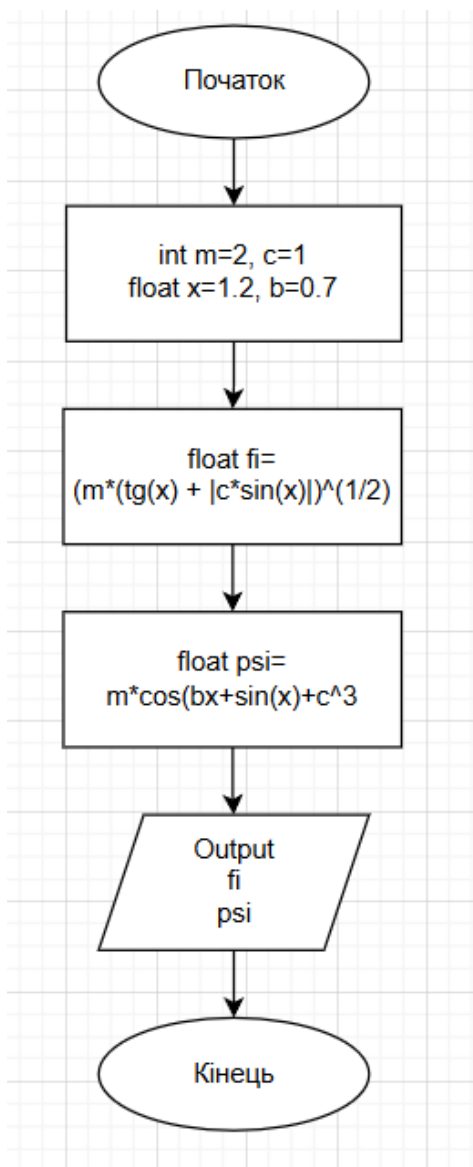
Розв'язок:

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int main() {
6      int m = 2, c = 1;
7      float x = 1.2, b = 0.7;
8      float fi = sqrt(m * tan(x) + abs(c * sin(x)));
9
10     float psi = m * cos(b * x + sin(x)) + pow(c, 3);
11
12     cout << "fi= " << fi << endl;
13     cout << "psi= " << psi << endl;
14 }
```

Результат виконання:

```
fi= 2.46502
psi= 0.600226
```

Блок-схема:



Завдання №2

VNS Practice Work Task 2

Варіант 16. Обчислити функцію $y = \begin{cases} \cos(ax + 2), & x > 2, \\ \operatorname{tg}|x - 2a|, & x \leq 2; \end{cases}$ де

$x \in [0, 5; 3, 1]$; $h_x = 0, 2$, a – має початкове значення 0,1 і змінюється одночасно зі змінною x з кроком $h_a = 0, 3$.

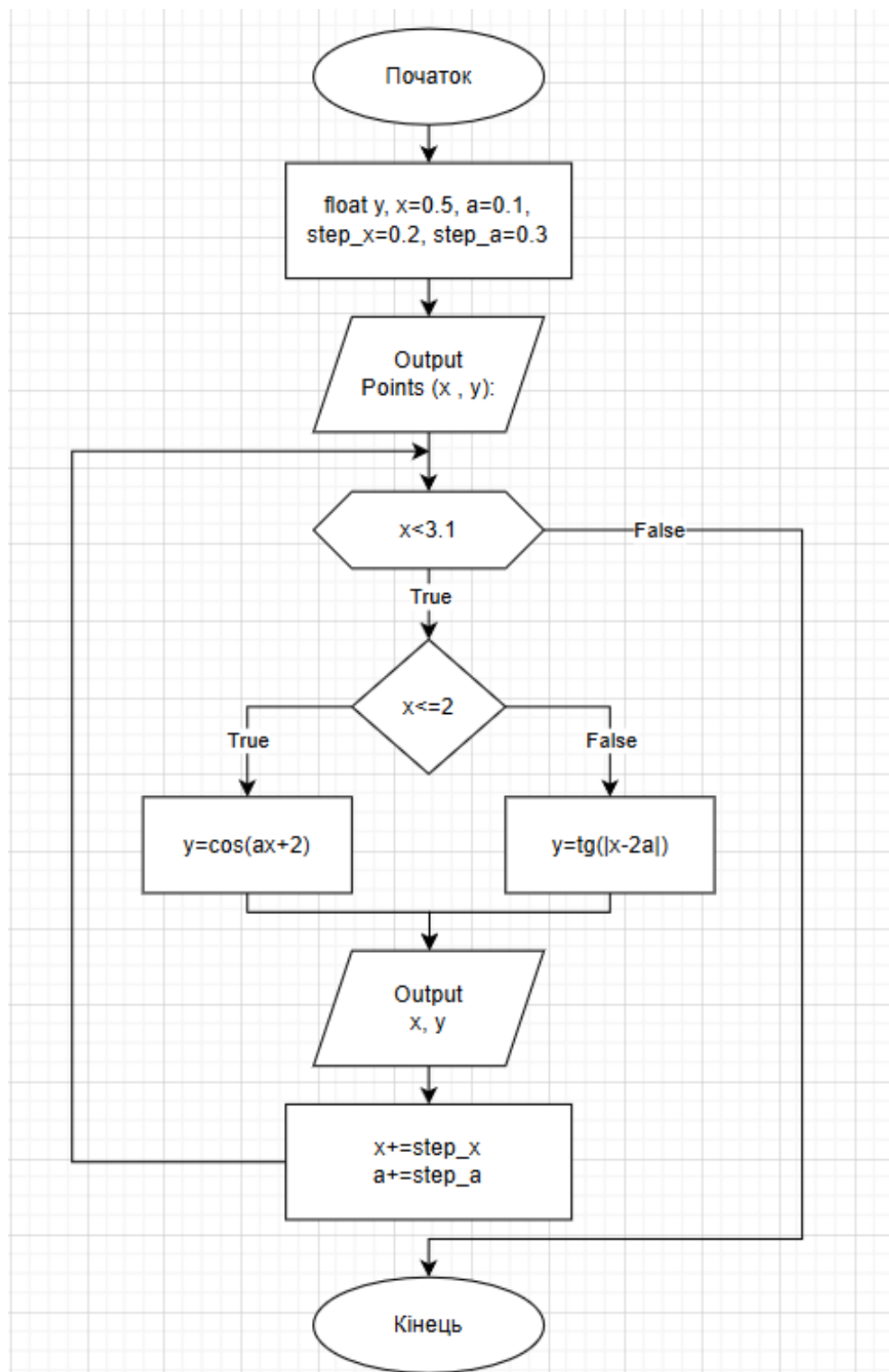
Розв'язок:

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int main() {
6      float x = 0.5, a = 0.1, step_x = 0.2, step_a = 0.3, y;
7      cout << "Points (x , y):" << endl;
8      while (x <= 3.1) {
9          if (x <= 2) {
10             y = cos(a * x + 2);
11         } else {
12             y = tan(abs(x - 2 * a));
13         }
14         printf("(%8.6f , %9.6f)\n", x, y);
15         x += step_x;
16         a += step_a;
17     }
18 }
```

Результат виконання:

```
Points (x , y):
(0.500000 , -0.461073)
(0.700000 , -0.651230)
(0.900000 , -0.871966)
(1.100000 , -0.999135)
(1.300000 , -0.853356)
(1.500000 , -0.307333)
(1.700000 , 0.494806)
(1.900000 , 0.994681)
(2.100000 , -0.246406)
(2.300000 , 0.159745)
(2.500000 , 0.624732)
(2.700000 , 1.423523)
(2.900000 , 4.637311)
```

Блок-схема:



Завдання №3

VNS Practice Work Task 3

Варіант 11. Обчислення вартості поїздки на автомобілі на дачу (туди і назад). Початковими даними є: відстань до дачі (км); кількість бензину, яка споживає автомобіль на 100 км пробігу; ціна одного літра бензину. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення вартості поїздки на дачу і назад.

Введіть початкові дані:

Відстань до дачі (км) > **67**

Витрата бензину (літрів на 100 км пробігу) > **8.5**

Ціна літра бензину (грн.) > **6.5**

Поїздка на дачу і назад обійдеться в 74.04 грн.

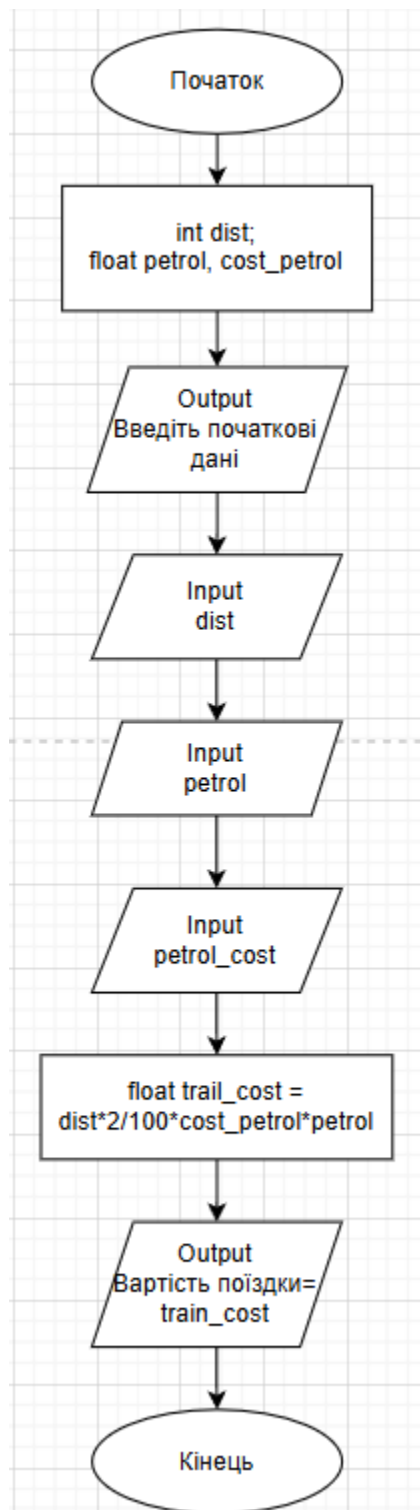
Розв'язок:

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int main() {
6      int dist;
7      float petrol, cost_petrol;
8      cout << "Введіть початкові дані:" << endl;
9      cout << "Відстань до дачі (км) > ";
10     cin >> dist;
11     cout << "Витрата бензину (літрів на 100 км пробігу) > ";
12     cin >> petrol;
13     cout << "Ціна літра бензину (грн.) > ";
14     cin >> cost_petrol;
15     float trail_cost = float(dist * 2) / 100 * cost_petrol * petrol;
16     printf("Поїздка на дачу і назад обійдеться в %.2f грн.\n", trail_cost);
17 }
```

Результат виконання:

```
Введіть початкові дані:
Відстань до дачі (км) > 67
Витрата бензину (літрів на 100 км пробігу) > 8.5
Ціна літра бензину (грн.) > 6.5
Поїздка на дачу і назад обійдеться в 74.04 грн.
```

Блок-схема:



Завдання №4

VNS Practice Work Task 4

Варіант 10. Написати програму, яка обчислює факторіал введеного з клавіатури числа.

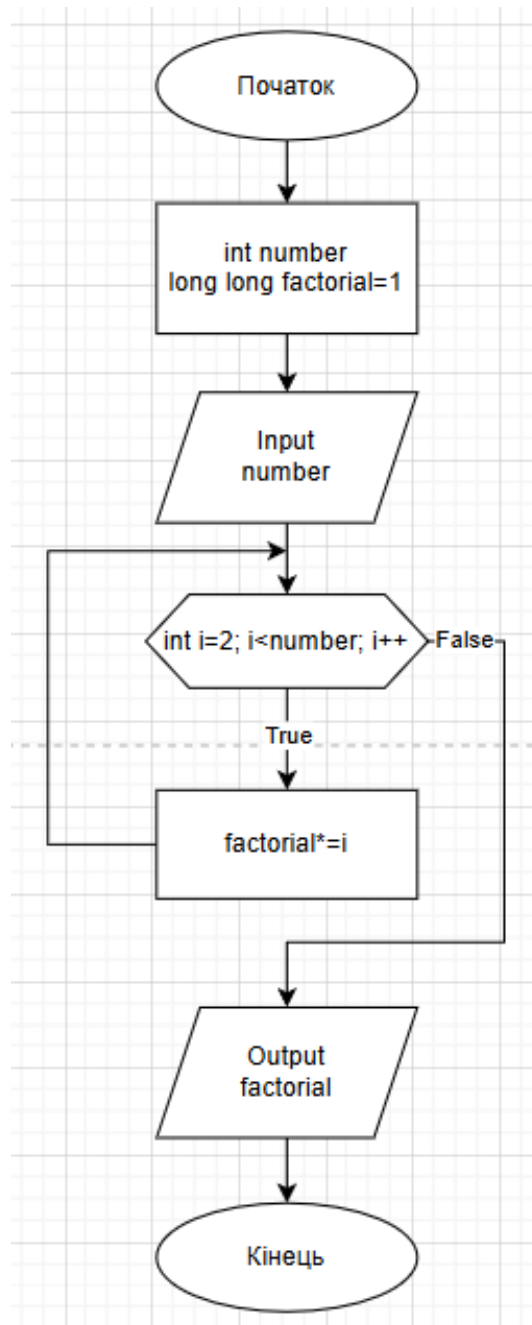
Розв'язок:

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int main() {
6      int number;
7      long long factorial=1;
8      cout<<"Number:";
9      cin>>number;
10     for(int i=2;i<=number;i++){
11         factorial*=i;
12     }
13     cout<<number<<"!= " <<factorial;
14 }
```

Результат виконання:

```
Number:9
9!= 362880
```


Блок-схема:



Завдання №5

Self-practice work Algotester №1

1225 Найкоротший шлях 2

Розв'язок:

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  //1225 Найкоротший шлях 2
5  int main() {
6      int n, m, x, y;
7      cin >> n >> m;
8
9      vector<vector<long long>> a;
10     long long inf = 1e14;
11     vector<vector<int>> grap(n);
12     long long l;
13     for (int i = 0; i < m; i++) {
14
15         cin >> x >> y >> l;
16         a.push_back({x - 1, y - 1, l});
17         grap[x - 1].push_back(y - 1);
18     }
19
20     vector<long long> d(n, inf);
21     d[0] = 0;
22     vector<bool> was(n, false);
23     was[0] = true;
24     for (int i = 0; i < n; i++) {
25         for (int j = 0; j < m; j++) {
26             int to = a[j][1];
27             int fro = a[j][0];
28             long long le = a[j][2];
29             if ((d[to] > d[fro] + le) && (was[fro])) {
30                 was[to] = true;
31                 d[to] = d[fro] + le;
32             }
33         }
34     }
35
36     queue<int> q;
37     for (int j = 0; j < m; j++) {
38         int to = a[j][1];
39         int fro = a[j][0];
40         long long le = a[j][2];
41         if ((d[to] > d[fro] + le) && (was[fro])) {
42             d[to] = min(d[to], d[fro] + le);
43             q.push(to);
44         }
45     }
46
47     vector<bool> visited(n, true);
48     while (!q.empty()) {
49         int cur = q.front();
50         q.pop();
51         d[cur] = -inf;
52         visited[cur] = false;
53         for (int i : grap[cur]) {
54             if (visited[i]) {
55                 visited[i] = false;
56                 q.push(i);
57             }
58         }
59     }
60
61     for (int i = 0; i < n; i++) {
62         if (d[i] == -inf) {
63             printf("-inf ");
64         } else if (d[i] == inf) {
65             printf("inf ");
66         } else {
67             printf("%lld ", d[i]);
68         }
69     }
70 }

```

Результат виконання:

```
8 9
1 2 4
2 3 7
3 4 -10
4 2 -3
3 5 5
1 6 -4
6 7 11
7 5 10
8 6 -5
0 -inf -inf -inf -inf -4 7 inf
```

місяць тому

C++ 23

Зараховано

0.216

2.117 [Перегляд](#)

Завдання №6

Self-practice work Algotester №2

0249 *Бійка у верховній раді*

Розв'язок:

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  //0249 Бійка у верховній раді
4  void read_agressive(int n,vector<int> &calm){
5      int x;
6      for(int i=0;i<n;i++){
7          cin>>x;
8          calm.push_back(x);
9      }
10 }
11
12 void read_friend(int m,vector<vector<int>> &g){
13     int x,y;
14     for(int i=0;i<m;i++){
15         cin>>x>>y;
16         x--;
17         y--;
18         g[x].push_back(y);
19         g[y].push_back(x);
20     }
21 }
22 void read_fighter(int k, vector<int> &calm, queue<int> &q){
23     int x;
24     for(int i=0;i<k;i++){
25         cin>>x;
26         x--;
27         calm[x]=0;
28         q.push(x);
29     }
30 }
31
32 int dfs(vector<int> &calm, vector<vector<int>> g, queue<int> &q){
33     int rez=0;
34     while(!q.empty()){
35         int cur = q.front();
36         q.pop();
37         for(auto v:g[cur]){
38             if (calm[v]==1){
39                 rez++;
40                 q.push(v);
41             }
42             calm[v]=max(calm[v]-1,0);
43         }
44     }
45     return rez;
46 }
47
48
49 int main() {
50     int n1, n2, m1, m2, k1, k2;
51     cin>>n1>>n2;
52     vector<vector<int>> g1(n1),g2(n2);
53     vector<int> calm1,calm2;
54     read_agressive(n1,calm1);
55     read_agressive(n2,calm2);
56     cin>>m1>>m2;
57     read_friend(m1,g1);
58     read_friend(m2,g2);
59     cin>>k1>>k2;
60     queue<int> q1, q2;
61     read_fighter(k1,calm1,q1);
62     read_fighter(k2,calm2,q2);
63
64
65     int result = dfs(calm1,g1,q1)+dfs(calm2,g2,q2)+k1+k2;
66     cout<<result;
67 }

```

Результат виконання:

```
6 3
47 7 2 1 4 2
1 2 1
6 2
1 2
1 3
2 3
3 4
3 5
4 6
1 2
2 3
2 1
2 5
3
5
```

7 днів тому

C++ 20

Зараховано

1.116

26.074 [Перегляд](#)

Завдання №7

Self-practice work Algotester №3

2133 Катеринка

Розв'язок:

```

1  #include<bits/stdc++.h>
2
3  using namespace std;
4  const int inf = 2e8;
5  //2133 Катеринка
6  void dijkstra(set<pair<int, int>> &que, const vector<vector<pair<int, int>>> &graph, vector<int> &dist) {
7      while (!que.empty()) {
8
9          int cur = que.begin()->second;
10         que.erase(que.begin());
11         for (auto curent: graph[cur]) {
12             if (dist[curent.first] > dist[cur] + curent.second) {
13                 que.erase({dist[curent.first], curent.first});
14                 dist[curent.first] = dist[cur] + curent.second;
15                 que.insert({dist[curent.first], curent.first});
16             }
17         }
18     }
19 }
20
21 int main() {
22     int n, m, q, kat, from, to, d;
23     cin >> n >> m >> q >> kat;
24     vector<int> dist(n, inf);
25     vector<vector<pair<int, int>>> graph(n);
26     set<pair<int, int>> que;
27     vector<pair<int, int>> find(q);
28     for (int i = 0; i < m; i++) {
29         cin >> from >> to >> d;
30         graph[from - 1].push_back({to - 1, d});
31         graph[to - 1].push_back({from - 1, d});
32     }
33     for (int i = 0; i < q; i++) {
34         cin >> find[i].first >> find[i].second;
35     }
36     kat--;
37     que.insert({0, kat});
38     dist[kat] = 0;
39     dijkstra(que, graph, dist);
40     for (int i = 0; i < q; i++) {
41         cout << dist[find[i].first - 1] + dist[find[i].second - 1] << endl;
42     }
43 }

```

Результат виконання:

```

4 5 3 1
1 2 1
2 3 3
3 1 1
3 4 4
1 4 10
4 3
2 3
1 4
6
2
5

```

місяць тому

C++ 23

Зараховано

0.707

31.766 [Перегляд](#)

Завдання №8

Self-practice work Algotester №4

1220 Каркас мінімальної ваги

Розв'язок:

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  vector<int>p={0};
4  vector<int>s={0};
5  //1220 Каркас мінімальної ваги
6  void init(int n){
7      for(int i=1;i<=n;i++){
8          p.push_back(i);
9          s.push_back(1);
10     }
11 }
12 int find(int x){
13     if(p[x]==x)
14         return p[x];
15     else
16         return p[x]=find(p[x]);
17 }
18
19 bool unite(int x, int y){
20     x=find(x);
21     y=find(y);
22     if(x==y) return false;
23     else{
24         if(s[x]>s[y]) {
25             s[x]+=s[y];
26             p[y] = x;
27         }
28         else{
29             s[y]+=s[x];
30             p[x]=y;
31         }
32     }
33     return true;
34 }
35 struct Edge{
36     int v;
37     int u;
38     int w;
39 };
40
41 int main(){
42     int n,m;
43     cin>>n>>m;
44     vector<Edge> graph(m+1);
45     for(int i=0;i<m;i++){
46         cin>>graph[i].v>>graph[i].u>>graph[i].w;
47     }
48     sort(graph.begin(),graph.end(), [&](Edge l, Edge r){
49         return l.w<r.w;
50     });
51     init(n);
52     long long rez=0;
53     for(Edge edge:graph){
54         if(unite(edge.v,edge.u)){
55             rez+=edge.w;
56         }
57     }
58     cout<<rez;
59 }

```

Результат виконання:

```
6 9
1 2 3
1 3 2
1 4 6
2 3 1
2 5 2
2 6 3
3 5 1
4 6 1
5 6 3
8
```

місяць тому	C++ 23	Зараховано	0.320	5.031	Перегляд
-------------	--------	------------	-------	-------	--------------------------

Висновок: в ході виконання розрахунково-графічної роботи я одержав практичні навички в розробці і дослідженні алгоритмів розв’язання задач.