

```

1  /*Se citeste din fisierul "arbore.txt" de pe prima linie
2  n - numarul de noduri si rad - radacina arborelui iar
3  de pe liniile urmatoare perechi i j care reprezinta
4  muchiile(ramurile) unui arbore A.
5  a. Pe baza matricei de adiacenta sa se construiasca si sa
6  se afiseze vectorul de tati
7  b. Sa se calculeze si sa se afiseze ordinul fiecarui nod.
8  c. Sa se afiseze frunzele arborelui.
9  */
10 /*Fisierul "arbore.txt"
11 12 6
12 1 6
13 2 6
14 4 6
15 1 5
16 1 7
17 9 4
18 4 8
19 4 10
20 8 11
21 8 12
22 5 3
23 */
24
25 #include <iostream>
26 #include <fstream>
27
28 using namespace std;
29
30 ifstream f("arbore.txt");
31
32 int n, rad, tata[30], a[30][30], ordine[30];
33 /*n -numarul de noduri
34 rad - radacina arborelui
35 tata - vectorul de tati
36 a - matricea de adiacenta
37 ordine - vectorul in care se retine ordinul fiecarui nod(numarul
38 de aparitii al nodului i in vectorul de tati
39 */
40
41 /* Functia Citire construiește matricea de adiacenta
42 si citeste radacina arborelui
43 */
44
45 void Citire()
46 {
47     int i, j;
48     f>>n>>rad;
49     while(f>>i>>j)
50         a[i][j]=a[j][i]=1;
51 }
52
53 /*Generarea vectorului de tati se realizeaza
54 printr-o parcurgere in latime a arborelui
55 plecand din radacina(BFS(rad)).
56 */
57
58 void GenVectTata(int rad)
59 {
60     int coada[30], viz[30], i, prim, ultim;
61     for(i=1;i<=n;i++)
62         coada[i]=viz[i]=0;
63     prim=ultim=1;
64     coada[1]=rad;
65     viz[rad]=1;
66     while (prim<=ultim)

```

```

67     {
68         for(i=1;i<=n;i++)
69             if(a[coada[prim]][i]==1 && viz[i]==0)
70                 {
71                     ultim++;
72                     coada[ultim]=i;
73                     viz[i]=1;
74                     tata[i]=coada[prim];
75                 }
76             }
77         prim++;
78     }
79 }
80
81 void AfisareVectTata()
82 {
83     cout<<"Vectorul de tati= ";
84     for(int i=1; i<=n; i++)
85         cout<<tata[i]<<" ";
86 }
87
88 void CalculOrdinNoduri()
89 {
90     int i, j;
91     for(i=1; i<=n; i++)
92         for(j=1; j<=n; j++)
93             if(tata[j]==i)
94                 ordine[i]++;
95 }
96
97 void AfisareOrdine()
98 {
99     int i;
100    cout<<"\nOrdinele nodurilor:";
101    for(i=1; i<=n; i++)
102        cout<<"\nOrdinul nodului " <<i<<"\t= " <<ordine[i];
103 }
104
105 void AfisareFrunze()
106 {
107     cout<<"\nFrunzele arborelui sunt: ";
108     for(int i=1; i<=n; i++)
109         if(ordine[i]==0)
110             cout<<" " <<i;
111 }
112 int main()
113 {
114     Citire();
115     GenVectTata(rad);
116     cout<<"Punctul a.\n";
117     AfisareVectTata();
118     CalculOrdinNoduri();
119     cout<<"\n\nPunctul b.";
120     AfisareOrdine();
121     cout<<"\n\nPunctul c.";
122     AfisareFrunze();
123     return 0;
124 }

```