

```

1 int invita(int N, int M, int conoscenzeA[], int conoscenzeB[]) {
2     std::vector<std::unordered_set<int>> adj(N);
3
4     for (int i = 0; i < M; ++i) {
5         int u = conoscenzeA[i];
6         int v = conoscenzeB[i];
7
8         adj[u].insert(v);
9         adj[v].insert(u);
10    }
11
12    int nodi_rimossi = 0;
13    // Cerchiamo tutti i nodi isolati
14    for (int i = 0; i < N; ++i)
15        if (adj[i].empty())
16            ++nodi_rimossi;
17
18    // Cerchiamo tutte le foglie del grafo iniziale
19    std::queue<int> foglie;
20    for (int i = 0; i < N; ++i)
21        if (adj[i].size() == 1)
22            foglie.push(i);
23
24    while (!foglie.empty()) {
25        int u = foglie.front();
26        foglie.pop();
27
28        // Rimuovi tutti gli archi da u ai vicini
29        for (const auto& v: adj[u]) {
30            adj[v].erase(u);
31
32            // Controllo se ho creato una nuova foglia
33            if (adj[v].size() == 1)
34                foglie.push(v);
35        }
36        adj[u].clear();
37
38        ++nodi_rimossi;
39    }
40
41    return N - nodi_rimossi;
42 }

```