{

    "": {

        "prefix": "graph\_trees\_height\_and\_depth",

        "body": [

          "const int M = 1e5+10;",

          "vector<int> g[M];",

          "int depth[M], height[M];",

          "",

          "",

          "void dfs\_for\_trees(int vertex, int parent)",

          "{",

          "    for(int child:g[vertex])",

          "    {",

          "        if(child==parent) continue;",

          "        depth[child]=depth[vertex] + 1;",

          "        dfs\_for\_trees(child, vertex);",

          "        height[vertex]=max(height[vertex], height[child] + 1);",

          "    }",

          "}",

          "",

          "",

          "void graph\_input()",

          "{",

          "    int n;",

          "    cin>>n;",

          "",

          "    for(int i=1; i<n; i++)",

          "    {",

          "        int x, y;",

          "        cin>>x>>y;",

          "        g[x].push\_back(y);",

          "        g[y].push\_back(x);",

          "    }",

          "",

          "    dfs\_for\_trees(1, -1);",

          "}",

          "",

          "void answer()",

          "{",

          "    graph\_input();",

          "}"

        ],

        "description": ""

      }

}