

In a plumbing network, water flows through unidirectional pipes that connect from one collection/distribution node to another. A node can have zero or more incoming and/or outgoing pipes. A plumbing network is effectively a directed graph. If water is accepted into a node from an incoming pipe, it is distributed onward to a selected subset of outgoing pipes, dividing the amount of water evenly among the selected outgoing pipes. In this problem, we are given a plumbing network that is completely dry, we dump a measured amount of water into one of the nodes—the *root* node, and compute how much water is accepted at each of the nodes. We will make several simplifying assumptions about how water flows:

- The pipes can handle whatever amount of water they are given. There is no such thing as restricted flow or backups; whenever a flow of water first reaches a node, it has already finished leaving the node from which it came.
- Any amount of water travels through a pipe from one node to another in a fixed amount of time; therefore, water flows outward breadth-first, reaching all nodes at distance one from the root simultaneously, all nodes at distance two from the root simultaneously thereafter, and so on.
- No water is distributed from a node to another node that has already accepted water; the outgoing pipes selected are those leading to nodes that are still dry.
- Water is accepted into a node from only one other node, and after it has been distributed onward, no further water is accepted. If water first reaches a node from two or more other nodes, only the water arriving from the lowest numbered other node is accepted.

### *Input Format*

Each node is represented by a positive integer. The first part of the input consists of nonempty lines containing an adjacency lists representation of a plumbing network, followed by an empty input line. Each line in the first part contains a node (in increasing order from line to line) followed by the other nodes (in increasing order) to which it is directly connected by outgoing pipes, separated by white space. The remaining input lines describe various scenarios in which some amount of water—a positive floating-point number—is initially dumped into some root node.

### *Output Format*

For each amount of water and root node in the second part of the input, compute the amount of water that will be accepted at each node in the initially dry plumbing network described in the first part of the input. At each node, report the amount of water accepted (if any), accurate to two decimal places, and the other node (if any) from which it was accepted. Format the output as shown in the sample on the next page.

### *Input Sample*

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

```
72.  1
98.6 3
100. 7
```

### *Output Sample*

```
1 accepts 72.00
2 accepts 36.00 from 1
3 accepts 12.00 from 4
4 accepts 36.00 from 1
5 accepts 12.00 from 4
6 accepts 12.00 from 4
7 accepts 18.00 from 8
8 accepts 18.00 from 2
9 accepts 18.00 from 2
10 accepts 6.00 from 3
11 accepts 6.00 from 3
```

```
1 accepts nothing
2 accepts 32.87 from 3
3 accepts 98.60
4 accepts nothing
5 accepts nothing
6 accepts nothing
7 accepts 16.43 from 8
8 accepts 16.43 from 2
9 accepts 16.43 from 2
10 accepts 32.87 from 3
11 accepts 32.87 from 3
```

```
1 accepts nothing
2 accepts nothing
3 accepts nothing
4 accepts nothing
5 accepts nothing
6 accepts nothing
7 accepts 100.00
8 accepts nothing
9 accepts nothing
10 accepts nothing
11 accepts nothing
```