

E. Beavermuncher-0xFF

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

"Eat a beaver, save a tree!" — That will be the motto of ecologists' urgent meeting in Beaverley Hills.

And the whole point is that the population of beavers on the Earth has reached incredible sizes! Each day their number increases in several times and they don't even realize how much their unhealthy obsession with trees harms the nature and the humankind. The amount of oxygen in the atmosphere has dropped to 17 per cent and, as the best minds of the world think, that is not the end.

In the middle of the 50-s of the previous century a group of soviet scientists succeed in foreseeing the situation with beavers and worked out a secret technology to clean territory. The technology bears a mysterious title "Beavermuncher-0xFF". Now the fate of the planet lies on the fragile shoulders of a small group of people who has dedicated their lives to science.

The prototype is ready, you now need to urgently carry out its experiments in practice.

You are given a tree, completely occupied by beavers. A tree is a connected undirected graph without cycles. The tree consists of n vertices, the i -th vertex contains k_i beavers.

"Beavermuncher-0xFF" works by the following principle: being at some vertex u , it can go to the vertex v , if they are connected by an edge, and eat **exactly one** beaver located at the vertex v . It is impossible to move to the vertex v if there are no beavers left in v . "Beavermuncher-0xFF" **cannot** just stand at some vertex and eat beavers in it. "Beavermuncher-0xFF" must move without stops.

Why does the "Beavermuncher-0xFF" works like this? Because the developers have not provided place for the battery in it and eating beavers is necessary for converting their mass into pure energy.

It is guaranteed that the beavers will be shocked by what is happening, which is why they will not be able to move from a vertex of the tree to another one. As for the "Beavermuncher-0xFF", it can move along each edge in both directions while conditions described above are fulfilled.

The root of the tree is located at the vertex s . This means that the "Beavermuncher-0xFF" begins its mission at the vertex s and it must return there at the end of experiment, because no one is going to take it down from a high place.

Determine the maximum number of beavers "Beavermuncher-0xFF" can eat and return to the starting vertex.

Input

The first line contains integer n — the number of vertices in the tree ($1 \leq n \leq 10^5$). The second line contains n integers k_i ($1 \leq k_i \leq 10^5$) — amounts of beavers on corresponding vertices. Following $n - 1$ lines describe the tree. Each line contains two integers separated by space. These integers represent two vertices connected by an edge. Vertices are numbered from 1 to n . The last line contains integer s — the number of the starting vertex ($1 \leq s \leq n$).

Output

Print the maximum number of beavers munched by the "Beavermuncher-0xFF".

Please, do not use `%lld` specifier to write 64-bit integers in C++. It is preferred to use `cout` (also you may use `%I64d`).

Examples

| input |
|-----------|
| 5 |
| 1 3 1 3 2 |
| 2 5 |

| |
|--------|
| 3 4 |
| 4 5 |
| 1 5 |
| 4 |
| output |
| 6 |

| |
|--------|
| input |
| 3 |
| 2 1 1 |
| 3 2 |
| 1 2 |
| 3 |
| output |
| 2 |