Lab 7 - Tree

#A

Description

Given the in-order and post-order of a binary tree. You need to find the pre-order.

Input

in-order.

post-order.

Output

pre-order.

Sample Test Data

Input #1

```
2143
2431
```

Output #1

```
1234
```

Tips:

```
   1
 /   \
2     3
     /
   4
```

Data Limit

Each node is represented by a single character. the character is a digit or a uppercase or lowercase letter. (i.e. [0-9a-zA-Z])

#B

Description

You are given a **tree** with N vertices.

An edge connects Vertices *a* and *b* and has a length of *c*.

You are also given *Q* queries and an integer *R*. In the *i*-th query:

find the length of the shortest path from Vertex x_i to Vertex y_i via Vertex *R*.

(Hint: what is the shortest path in a tree? Don't think it's complicated.)

Input

N

N−1 lines, each line is: *a b c*

Q R

Q lines, each line is: x_i y_i

Output

Q lines, each line is a length of the shortest path.

Sample Test Data

Input #1

```
5
1 2 1
1 3 1
2 4 1
3 5 1
3 1
2 4
2 3
4 5
```

Output #1

```
3
2
4
```

Input #2

```
10
1 2 1000000000
2 3 1000000000
3 4 1000000000
4 5 1000000000
5 6 1000000000
6 7 1000000000
7 8 1000000000
8 9 1000000000
9 10 1000000000
1 1
9 10
```

Output #2

```
17000000000
```

Input #3

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

Output #3

```
5
14
22
```

Data Limit

$3 \le N \le 100000$

$1 \le Q \le 100000$

$x_i \neq y_i, x_i \neq R, y_i \neq R$

$1 \le c \le 10^9$

#C

Description

You are given a **tree** with N vertices.

An edge connects Vertices *a* and *b* and has a length of 1.

You need to find a vertex such that the sum of the shortest distances from all vertices to it is the minimum.

If multiple vertices meet the requirements, use the one with the lowest number.

(Hint: the vertex is called "centroid". It has some properties. You can easily find more information about it online.)

Input

N

N−1 lines, each line is: *a* *b*, representing an edge.

Output

A line, two integers.

The first is the vertex.

The second is the sum of the shortest distances from all vertices to it.

Sample Test Data

Input #1

```
4
1 2
2 3
3 4
```

Output #1

```
2 4
```

Data Limit

$2 \leq N \leq 100000$

#D

Description

A text has *n* kinds of characters, you need to encode the characters in binary. The encoding of one kind of character cannot be a prefix of another.

The number of occurrences of the *i*-th character is a_i how to make its final text encoding length minimum. In the case of ensuring the minimum length of the final text encoding, how to make the longest character encoding minimum ?

Input

n

n lines, each line: a_i

Output

The minimum length of the text after encoding.

In the case of ensuring the minimum length of the final encoding, the minimum longest character encoding.

Sample Test Data

Input #1

```
6
1
5
7
9
1
6
```

Output #1

```
67
4
```

Data Limit

$1 \le n \le 100000$

$1 \le a_i \le 10^9$

Hint

`priority_queue` in C++ or `PriorityQueue` in Java may help you.

#E

Description

You are given a **tree** with N vertices.

Each vertex has an initial value 0.

An edge e_i connects Vertices a_{e_i} and b_{e_i}.

You are also given *Q* queries. In the *i*-th query:

You are given integers: *opt*, e_i, *x*.

if *opt*=1: for each Vertex *v* reachable from Vertex a_{e_i} without visiting Vertex b_{e_i} by traversing edges, the value of *v* increases by *x*.

if *opt*=2: for each Vertex *v* reachable from Vertex b_{e_i} without visiting Vertex a_{e_i} by traversing edges, the value of *v* increases by *x*.

Input

N

N−1 lines, each line is: a_{e_i} b_{e_i}

Q

Q lines, each line is: *opt* e_i *x*

Output

The value of each vertex at the end, in the order of 1~n vertex.

Sample Test Data

Input #1

```
5
1 2
2 3
2 4
4 5
4
1 1 1
1 4 10
2 1 100
2 2 1000
```

Output #1

```
11
110
1110
110
100
```

Tips

```
     1
    /e1
   2
 /e2 \e3
3     4
       \e4
        5
```

In the first query, we add 1 to Vertex 1.

In the second query, we add 10 to Vertex 1,2,3,4

In the third query, we add 100 to Vertex 2,3,4,5.

In the fourth query, we add 1000 to Vertex 3.

Input #2

```
7
2 1
2 3
4 2
4 5
6 1
3 7
7
2 2 1
1 3 2
2 2 4
1 6 8
1 3 16
2 4 32
2 1 64
```

Output #2

```
72
8
13
26
58
72
5
```

Input #3

```
11
2 1
1 3
3 4
5 2
1 6
```

```
1 7
5 8
3 9
3 10
11 4
10
2 6 688
1 10 856
1 8 680
1 8 182
2 2 452
2 4 183
2 6 518
1 3 612
2 6 339
2 3 206
```

output #3

```
1657
1657
2109
1703
1474
1657
3202
1474
1247
2109
2559
```

Data Limit

$2 \leq N \leq 200000$

$1 \leq Q \leq 200000$

$1 \leq x \leq 10^9$

Hint

Maybe you can save time by making some markers to tell you that an adding was performed instead of actually doing it.