A. Bear and Poker

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Limak is an old brown bear. He often plays poker with his friends. Today they went to a casino. There are n players (including Limak himself) and right now all of them have bids on the table. i-th of them has bid with size a_i dollars.

Each player can double his bid any number of times and triple his bid any number of times. The casino has a great jackpot for making all bids equal. Is it possible that Limak and his friends will win a jackpot?

Input

First line of input contains an integer n ($2 \leq n \leq 10^5$), the number of players.

The second line contains n integer numbers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$) — the bids of players.

Output

Print "Yes" (without the quotes) if players can make their bids become equal, or "No" otherwise.

Examples

input

```
4
75 150 75 50
```

output

```
Yes
```

input

```
3
100 150 250
```

output

```
No
```

Note

In the first sample test first and third players should double their bids twice, second player should double his bid once and fourth player should both double and triple his bid.

It can be shown that in the second sample test there is no way to make all bids equal.

B. Bear and Blocks

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Limak is a little bear who loves to play. Today he is playing by destroying block towers. He built n towers in a row. The i-th tower is made of h_i identical blocks. For clarification see picture for the first sample.

Limak will repeat the following operation till everything is destroyed.

Block is called internal if it has all four neighbors, i.e. it has each side (top, left, down and right) adjacent to other block or to the floor. Otherwise, block is boundary. In one operation Limak destroys all boundary blocks. His paws are very fast and he destroys all those blocks at the same time.

Limak is ready to start. You task is to count how many operations will it take him to destroy all towers.

Input

The first line contains single integer n ($1 \leq n \leq 10^5$).

The second line contains n space-separated integers $h_1, h_2, \ldots, h_n$ ($1 \leq h_i \leq 10^9$) — sizes of towers.

Output

Print the number of operations needed to destroy all towers.

Examples

input

```
6
2 1 4 6 2 2
```

output

```
3
```

input

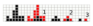
```
7
3 3 3 1 3 3 3
```

output

```
2
```

Note

The picture below shows all three operations for the first sample test. Each time boundary blocks are marked with red color.



After first operation there are four blocks left and only one remains after second operation. This last block is destroyed in third operation.

C. Bear and Drawing

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Limak is a little bear who learns to draw. People usually start with houses, fences and flowers but why would bears do it? Limak lives in the forest and he decides to draw a tree.

Recall that *tree* is a connected graph consisting of n vertices and $n - 1$ edges.

Limak chose a tree with n vertices. He has infinite strip of paper with two parallel rows of dots. Little bear wants to assign vertices of a tree to some n distinct dots on a paper so that edges would intersect only at their endpoints — drawn tree must be planar. Below you can see one of correct drawings for the first sample test.

Is it possible for Limak to draw chosen tree?

Input

The first line contains single integer n ($1 \le n \le 10^5$).

Next $n - 1$ lines contain description of a tree. i-th of them contains two space-separated integers a_i and b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$) denoting an edge between vertices a_i and b_i. It's guaranteed that given description forms a tree.

Output

Print "Yes" (without the quotes) if Limak can draw chosen tree. Otherwise, print "No" (without the quotes).

Examples

input

```
8
1 2
1 3
1 6
6 4
6 7
6 5
7 8
```

output

```
Yes
```

input

```
13
1 2
1 3
1 4
2 5
2 6
2 7
3 8
3 9
3 10
4 11
4 12
4 13
```

output

```
No
```

D. Bear and Cavalry

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

Would you want to fight against bears riding horses? Me neither.

Limak is a grizzly bear. He is general of the dreadful army of Bearland. The most important part of an army is cavalry of course.

Cavalry of Bearland consists of n warriors and n horses. i-th warrior has strength w_i and i-th horse has strength h_i. Warrior together with his horse is called a unit. Strength of a unit is equal to multiplied strengths of warrior and horse. Total strength of cavalry is equal to sum of strengths of all n units. Good assignment of warriors and horses makes cavalry truly powerful.

Initially, i-th warrior has i-th horse. You are given q queries. In each query two warriors swap their horses with each other.

General Limak must be ready for every possible situation. What if warriors weren't allowed to ride their own horses? After each query find the maximum possible strength of cavalry if we consider assignments of all warriors to all horses that no warrior is assigned to his own horse (it can be proven that for $n \geq 2$ there is always at least one correct assignment).

Note that we can't leave a warrior without a horse.

Input

The first line contains two space-separated integers, n and q ($2 \leq n \leq 30\,000$, $1 \leq q \leq 10\,000$).

The second line contains n space-separated integers, $w_1, w_2, \dots, w_n$ ($1 \leq w_i \leq 10^6$) — strengths of warriors.

The third line contains n space-separated integers, $h_1, h_2, \dots, h_n$ ($1 \leq h_i \leq 10^6$) — strengths of horses.

Next q lines describe queries. i-th of them contains two space-separated integers a_i and b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), indices of warriors who swap their horses with each other.

Output

Print q lines with answers to queries. In i-th line print the maximum possible strength of cavalry after first i queries.

Examples

input

```
4 2
1 10 100 1000
3 7 2 5
2 4
2 4
```

output

```
5732
7532
```

input

```
3 3
7 11 5
3 2 1
1 2
1 3
2 3
```

output

```
44
48
52
```

input

```
7 4
1 2 4 8 16 32 64
87 40 77 29 50 11 18
1 5
2 7
6 2
5 6
```

output

```
9315
9308
9315
9315
```

Note

Clarification for the first sample:

```
Warriors: 1 10 100 1000
Horses:   3  7  2    5
```

After first query situation looks like the following:

```
Warriors: 1 10 100 1000
Horses:   3  5  2    7
```

We can get $1 \cdot 2+10 \cdot 3+100 \cdot 7+1000 \cdot 5=5732$ (note that no hussar takes his own horse in this assignment).

After second query we get back to initial situation and optimal assignment is $1 \cdot 2+10 \cdot 3+100 \cdot 5+1000 \cdot 7=7532$.

Clarification for the second sample. After first query:

```
Warriors:  7 11 5
Horses:    2  3 1
```

Optimal assignment is $7 \cdot 1+11 \cdot 2+5 \cdot 3=44$.

Then after second query $7 \cdot 3+11 \cdot 2+5 \cdot 1=48$.

Finally $7 \cdot 2+11 \cdot 3+5 \cdot 1=52$.

E. Bear and Bowling

time limit per test: 6 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Limak is an old brown bear. He often goes bowling with his friends. Today he feels really good and tries to beat his own record!

For rolling a ball one gets a score — an integer (maybe negative) number of points. Score for i-th roll is multiplied by i and scores are summed up. So, for k rolls with scores $s_1, s_2, \dots, s_k$, total score is $\sum_{i=1}^{k} i \cdot s_i$. Total score is 0 if there were no rolls.

Limak made n rolls and got score a_i for i-th of them. He wants to maximize his total score and he came up with an interesting idea. He will cancel some rolls, saying that something distracted him or there was a strong wind.

Limak is able to cancel any number of rolls, maybe even all or none of them. Total score is calculated as if there were only non-canceled rolls. Look at the sample tests for clarification. What maximum total score can Limak get?

Input

The first line contains single integer n ($1 \le n \le 10^5$).

The second line contains n space-separated integers $a_1, a_2, \dots, a_n$ ($|a_i| \le 10^7$) - scores for Limak's rolls.

Output

Print the maximum possible total score after choosing rolls to cancel.

Examples

input

```
5
-2 -8 0 5 -3
```

output

```
13
```

input

```
6
-10 20 -30 40 -50 60
```

output

```
400
```

Note

In first sample Limak should cancel rolls with scores -8 and -3. Then he is left with three rolls with scores $-2, 0, 5$. Total score is $1 \cdot (-2) + 2 \cdot 0 + 3 \cdot 5 = 13$.

In second sample Limak should cancel roll with score -50. Total score is $1 \cdot (-10) + 2 \cdot 20 + 3 \cdot (-30) + 4 \cdot 40 + 5 \cdot 60 = 400$.