A Appeal to the Audience

You are the director of the upcoming Bordfite Arena Progaming Competition. You have invited a bunch of players and are now setting up the matches for the knockout tournament that will determine the winner. Asyoumayknow,BordfiteArenaisagamethatheavily rewards skill and very little luck is involved. This implies that whenever any number of players play a game of Bordfite Arena, the most skilled player will always win! Hence the winner of the tournament is already known, and you are a bit worried about this. How will you appease the audience? You embark on a short quest to find out what the audience finds interesting. No surprises there: people find it most interesting when they see skillful players compete. Whenever a match is played, the happiness the audience gets from a match is the sum of the skill levels of the players. The total happiness the audience gets from the tournament is the sum of the happiness obtained during all matches. This is very useful information, because of course you want the audience to be as happy as possible at the end of the tournament. Moreover, you invested some time to ask people what kind of knockout format they like. It turns out that instead of the plain old binary tree for the knockout schedule, they prefer a specific weird-looking rooted tree, and so you decide to use that. This means the final step for you to complete is to divide the players over the leaves of the given tree so that over the entire tournament, the happiness of the audience is maximized.

Input

• The first line contains integers $3 \leq n \leq 10^5$ and $1 \leq k \leq n-1$, the number of nodes of the tree and the number of players. The nodes are labelled 0 through $n-1$, and 0 is the root of the tree.

• The second line contains k integers $0 \leq a_1,...,a_k \leq 10^9$, denoting the skill values of the players.

• Then follow $n-1$ lines, the ith of which ($1 \leq i \leq n-1$) contains the parent $0 \leq p_i < i$ of node i.

It is guaranteed that the tree has exactly k leaves and that there are no nodes with exactly one child.

Output

• Output the maximal possible happiness the audience can obtain from this tournament.

Sample Input

5 3
5 4 3
0
0
1
1

Sample Output

17

B Deck Randomisation

Alice and Bob love playing Don’tminion, which typically involves a lot of shuffling of decks of diff erent sizes. Because theyplaysooften,theyarenotonlyveryquickatshuffling, but also very consistent. Each time Alice shuffles her deck, her cards get permuted in the same way, just like Bob always permutes his cards the same way when he shuffles them. This isn’t good for playing games, but raises an interesting question. They know that if they take turns shuffling, then at some point the deck will end up ordered in the same way as when they started. Alice shuffles once first, then Bob shuffles once, then Alice shuffles again, et cetera. They start with a sorted deck. What they do not know, however, is how many shuffles it will take before the deck is sorted again. Can you help them compute how many shuffles it will take? As Alice and Bob can only do 1012 shuffles in the limited time they have, any number strictly larger than this should be returned as huge instead.

Input

• The first line contains a single integer $1 \leq n \leq 10^5$, the number of cards in the deck.

• The second line contains n distinct integers $1 \leq a_1, a_2, \ldots, a_n \leq n$, where a_i is the new position of the card previously at position i when Alice shuffles the deck.

• The third line contains n distinct integers $1 \leq b_1, b_2, \ldots, b_n \leq n$, where b_i is the new position of the card previously at position i when Bob shuffles the deck.

Output

• Output a single positive integer $m > 0$, the minimal number of shuffles required to sort the deck, or huge when this number is strictly larger than 10^{12}.

Sample Input 1

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

Sample Output 1

```
2
```

Sample Input 2

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

Sample Output 2

```
5
```

C Efficient Exchange

You have recently acquired a new job at the Bank for Acquiring Peculiar Currencies. Here people can make payments, and deposit or withdraw money in all kinds of strange currencies. At your first day on the job you help a customer from Nijmegia, a small insignificant country famous for its enormous coins with values equal to powers of 10, that is, 1,10,100,1000, etc. This customer wants to make a rather large payment, and you are not looking forward to the prospect of carrying all those coins to and from the vault. You therefore decide to think things over first. You have an enormous supply of Nijmegian coins in reserve, as does the customer (most citizens from Nijmegia are extremely strong). You now want to minimize the total number of coins that are exchanged, in either direction, to make the exact payment the customer has to make. For example, if the customer wants to pay 83 coins there are many ways to make the exchange. Here are three possibilities:

Option 1. The customer pays 8 coins of value 10, and 3 coins of value 1. This requires exchanging 8+3=11 coins.

Option 2. The customer pays a coin of value 100, and you return a coin of value 10, and 7 coins of value 1. This requires exchanging 1+1+7=9 coins.

Option 3. The customer pays a coin of value 100, and 3 coins of value 1. You return 2 coins of value 10. This requires exchanging 1+3+2=6 coins.

It turns out the last way of doing it requires the least coins possible.

Input

• A single integer $0 \leq n < 10^{1000}$, the amount the customer from Nijmegia has to pay.

Output

• Output the minimum number of coins that have to be exchanged to make the required payment.

Sample Input

12345678987654321

Sample Output

42

D Jazz it Up!

Along with some friends you formed the Band of Atonal Percussionists and Cellists. You have been playing for some years together, but you feel unsatisfied with the current level of play. Doing research into some interesting new styles, you are gripped by the intricate details of the world of jazz. While of course you cannot apply all the new things you have learned immediately, you want to start with improvising some nice new rhythmic figures in the music your band plays. You will play a rhythm where every bar has n beats in it, but then you split up every beat into m notes. In total, you will have nm notes per bar. Everyone in the band knows that there is no room for squares in jazz. So the number of notes in a bar should be squarefree. That is, there is no number $k > 1$ such that k^2 divides the number of notes in a bar. The percussionist has already suggested a number of beats per bar n; now it is up to you to find a number of notes per beat that does not leave any room for squares. In the second sample we have $n = 30$ and $m = 7$. This works because $2 \leq m < n$ and $m \times n = 210$ has no divisor k^2 for any $k > 1$.

Input

• The input is a single squarefree integer $3 \leq n \leq 10^5$.

Output

• Output an integer $2 \leq m < n$ such that $m \times n$ is still squarefree.

If there are multiple possible solutions, you should output the minimal of them.

Sample Input

30

Sample Output

7