A. Primal Sport

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice and Bob begin their day with a quick game. They first choose a starting number $X_0 \geq 3$ and try to reach one million by the process described below.

Alice goes first and then they take alternating turns. In the i-th turn, the player whose turn it is selects a prime number smaller than the current number, and announces the smallest multiple of this prime number that is not smaller than the current number.

Formally, he or she selects a prime $p < X_{i-1}$ and then finds the minimum $X_i \geq X_{i-1}$ such that p divides X_i. Note that if the selected prime p already divides X_{i-1}, then the number does not change.

Eve has witnessed the state of the game after two turns. Given X_2, help her determine what is the smallest possible starting number X_0. Note that the players don't necessarily play optimally. You should consider all possible game evolutions.

Input

The input contains a single integer X_2 ($4 \leq X_2 \leq 10^6$). It is guaranteed that the integer X_2 is composite, that is, is not prime.

Output

Output a single integer — the minimum possible X_0.

Examples

input
```
14
```
output
```
6
```

input
```
20
```
output
```
15
```

input
```
8192
```
output
```
8191
```

Note

In the first test, the smallest possible starting number is $X_0 = 6$. One possible course of the game is as follows:

- Alice picks prime 5 and announces $X_1 = 10$
- Bob picks prime 7 and announces $X_2 = 14$.

In the second case, let $X_0 = 15$.

- Alice picks prime 2 and announces $X_1 = 16$
- Bob picks prime 5 and announces $X_2 = 20$.

B. Producing Snow

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

Alice likes snow a lot! Unfortunately, this year's winter is already over, and she can't expect to have any more of it. Bob has thus bought her a gift — a large snow maker. He plans to make some amount of snow every day. On day i he will make a pile of snow of volume V_i and put it in her garden.

Each day, every pile will shrink a little due to melting. More precisely, when the temperature on a given day is T_i, each pile will reduce its volume by T_i. If this would reduce the volume of a pile to or below zero, it disappears forever. All snow piles are independent of each other.

Note that the pile made on day i already loses part of its volume on the same day. In an extreme case, this may mean that there are no piles left at the end of a particular day.

You are given the initial pile sizes and the temperature on each day. Determine the total volume of snow melted on each day.

Input

The first line contains a single integer N ($1 \le N \le 10^5$) — the number of days.

The second line contains N integers $V_1, V_2, \ldots, V_N$ ($0 \le V_i \le 10^9$), where V_i is the initial size of a snow pile made on the day i.

The third line contains N integers $T_1, T_2, \ldots, T_N$ ($0 \le T_i \le 10^9$), where T_i is the temperature on the day i.

Output

Output a single line with N integers, where the i-th integer represents the total volume of snow melted on day i.

Examples

input

```
3
10 10 5
5 7 2
```

output

```
5 12 4
```

input

```
5
30 25 20 15 10
9 10 12 4 13
```

output

```
9 20 35 11 25
```

Note

In the first sample, Bob first makes a snow pile of volume 10, which melts to the size of 5 on the same day. On the second day, he makes another pile of size 10. Since it is a bit warmer than the day before, the first pile disappears completely while the second pile shrinks to 3. At the end of the second day, he has only a single pile of size 3. On the third day he makes a smaller pile than usual, but as the temperature dropped too, both piles survive till the end of the day.

C. Perfect Security

time limit per test: 3.5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Alice has a very important message M consisting of some non-negative integers that she wants to keep secret from Eve. Alice knows that the only theoretically secure cipher is one-time pad. Alice generates a random key K of the length equal to the message's length. Alice computes the bitwise xor of each element of the message and the key ($A_i := M_i \oplus K_i$, where $\oplus$ denotes the bitwise XOR operation) and stores this encrypted message A. Alice is smart. Be like Alice.

For example, Alice may have wanted to store a message $M = (0, 15, 9, 18)$. She generated a key $K = (16, 7, 6, 3)$. The encrypted message is thus $A = (16, 8, 15, 17)$.

Alice realised that she cannot store the key with the encrypted message. Alice sent her key K to Bob and deleted her own copy. Alice is smart. Really, be like Alice.

Bob realised that the encrypted message is only secure as long as the key is secret. Bob thus randomly permuted the key before storing it. Bob thinks that this way, even if Eve gets both the encrypted message and the key, she will not be able to read the message. Bob is not smart. Don't be like Bob.

In the above example, Bob may have, for instance, selected a permutation $(3, 4, 1, 2)$ and stored the permuted key $P = (6, 3, 16, 7)$.

One year has passed and Alice wants to decrypt her message. Only now Bob has realised that this is impossible. As he has permuted the key randomly, the message is lost forever. Did we mention that Bob isn't smart?

Bob wants to salvage at least some information from the message. Since he is not so smart, he asks for your help. You know the encrypted message A and the permuted key P. What is the lexicographically smallest message that could have resulted in the given encrypted text?

More precisely, for given A and P, find the lexicographically smallest message O, for which there exists a permutation π such that $O_i \oplus \pi(P_i) = A_i$ for every i.

Note that the sequence S is lexicographically smaller than the sequence T, if there is an index i such that $S_i < T_i$ and for all $j < i$ the condition

$S_j = T_j$ holds.

Input

The first line contains a single integer N ($1 \le N \le 300000$), the length of the message.

The second line contains N integers $A_1, A_2, \ldots, A_N$ ($0 \le A_i < 2^{30}$) representing the encrypted message.

The third line contains N integers $P_1, P_2, \ldots, P_N$ ($0 \le P_i < 2^{30}$) representing the permuted encryption key.

Output

Output a single line with N integers, the lexicographically smallest possible message O. Note that all its elements should be non-negative.

Examples

input

```
3
8 4 13
17 2 7
```

output

```
10 3 28
```

input

```
5
12 7 87 22 11
18 39 9 12 16
```

output

```
0 14 69 6 44
```

input

```
10
331415699 278745619 998190004 423175621 42983144 166555524 843586353 802130100 337889448 685310951
226011312 266003835 342809544 504667531 529814910 684873393 817026985 844010788 993949858 1031395667
```

output

```
128965467 243912600 4281110 112029883 223689619 76924724 429589 119397893 613490433 362863284
```

Note

In the first case, the solution is $(10, 3, 28)$, since $8 \oplus 2 = 10$, $4 \oplus 7 = 3$ and $13 \oplus 17 = 28$. Other possible permutations of key yield messages $(25, 6, 10)$, $(25, 3, 15)$, $(10, 21, 10)$, $(15, 21, 15)$ and $(15, 6, 28)$, which are all lexicographically larger than the solution.

D. Picking Strings

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

Alice has a string consisting of characters 'A', 'B' and 'C'. Bob can use the following transitions on any substring of our string in any order any number of times:

- $\mathtt{A} \longrightarrow \mathtt{BC}$
- $\mathtt{B} \longrightarrow \mathtt{AC}$
- $\mathtt{C} \longrightarrow \mathtt{AB}$
- $\mathtt{AAA} \longrightarrow$ empty string

Note that a substring is one or more consecutive characters. For given queries, determine whether it is possible to obtain the target string from source.

Input

The first line contains a string S ($1 \le |S| \le 10^5$). The second line contains a string T ($1 \le |T| \le 10^5$), each of these strings consists only of uppercase English letters 'A', 'B' and 'C'.

The third line contains the number of queries Q ($1 \le Q \le 10^5$).

The following Q lines describe queries. The i-th of these lines contains four space separated integers a_i, b_i, c_i, d_i. These represent the i-th query: is it possible to create $T[c_i..d_i]$ from $S[a_i..b_i]$ by applying the above transitions finite amount of times?

Here, $U[x..y]$ is a substring of U that begins at index x (indexed from 1) and ends at index y. In particular, $U[1..|U|]$ is the whole string U.

It is guaranteed that $1 \le a \le b \le |S|$ and $1 \le c \le d \le |T|$.

Output

Print a string of Q characters, where the i-th character is '1' if the answer to the i-th query is positive, and '0' otherwise.

Example

input

```
AABCCBAAB
ABCB
5
1 3 1 2
2 2 2 4
7 9 1 1
3 4 2 3
4 5 1 3
```

output

```
10011
```

Note

In the first query we can achieve the result, for instance, by using transitions $AA\mathbf{B} \to AA\mathbf{AC} \to \mathbf{AAA}AB \to AB$.

The third query asks for changing `AAB` to `A` — but in this case we are not able to get rid of the character '`B`'.

E. Perpetual Subtraction

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

There is a number x initially written on a blackboard. You repeat the following action a fixed amount of times:

1. take the number x currently written on a blackboard and erase it
2. select an integer uniformly at random from the range $[0, x]$ inclusive, and write it on the blackboard

Determine the distribution of final number given the distribution of initial number and the number of steps.

Input

The first line contains two integers, N ($1 \leq N \leq 10^5$) — the maximum number written on the blackboard — and M ($0 \leq M \leq 10^{18}$) — the number of steps to perform.

The second line contains $N + 1$ integers $P_0, P_1, \ldots, P_N$ ($0 \leq P_i < 998244353$), where P_i describes the probability that the starting number is i. We can express this probability as irreducible fraction P / Q, then $P_i \equiv PQ^{-1} \pmod{998244353}$. It is guaranteed that the sum of all P_is equals 1 (modulo 998244353).

Output

Output a single line of $N + 1$ integers, where R_i is the probability that the final number after M steps is i. It can be proven that the probability may always be expressed as an irreducible fraction P / Q. You are asked to output $R_i \equiv PQ^{-1} \pmod{998244353}$.

Examples

input

```
2 1
0 0 1
```

output

```
332748118 332748118 332748118
```

input

```
2 2
0 0 1
```

output

```
942786334 610038216 443664157
```

input

```
9 350
3 31 314 3141 31415 314159 3141592 31415926 314159265 649178508
```

output

```
822986014 12998613 84959018 728107923 939229297 935516344 27254497 413831286 583600448 442738326
```

Note

In the first case, we start with number 2. After one step, it will be 0, 1 or 2 with probability 1/3 each.

In the second case, the number will remain 2 with probability 1/9. With probability 1/9 it stays 2 in the first round and changes to 1 in the next, and with probability 1/6 changes to 1 in the first round and stays in the second. In all other cases the final integer is 0.

F. Public Service

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are N cities in Bob's country connected by roads. Some pairs of cities are connected by public transport. There are two competing transport companies — **Boblines** operating buses and **Bobrail** running trains. When traveling from A to B, a passenger always first selects the mode of transport (either bus or train), and then embarks on a journey. For every pair of cities, there are exactly two ways of how to travel between them without visiting any city more than once — one using only bus routes, and the second using only train routes. Furthermore, there is no pair of cities that is directly connected by both a bus route and a train route.

You obtained the plans of each of the networks. Unfortunately, each of the companies uses different names for the same cities. More precisely, the bus company numbers the cities using integers from 1 to N, while the train company uses integers between $N+1$ and $2N$. Find one possible mapping between those two numbering schemes, such that no pair of cities is connected directly by both a bus route and a train route. Note that this mapping has to map different cities to different cities.

Input

The first line contains an integer N ($2 \le N \le 10000$), the number of cities.

$N - 1$ lines follow, representing the network plan of Boblines. Each contains two integers u and v ($1 \le u, v \le N$), meaning that there is a bus route between cities u and v.

$N - 1$ lines follow, representing the network plan of Bobrail. Each contains two integers u and v ($N+1 \le u, v \le 2N$), meaning that there is a train route between cities u and v.

Output

If there is no solution, output a single line with the word "`No`".

If a solution exists, output two lines. On the first line, there should be the word "`Yes`". On the second line, there should be N integers $P_1, P_2, \ldots, P_N$ ($N+1 \le P_i \le 2N$) — the mapping between the two numbering schemes. More precisely, for $i \ne j$ it should be $P_i \ne P_j$, and for every direct bus route (i, j), there is no direct train route between (P_i, P_j).

If there are multiple solutions, you may print any of them.

Examples

input
```
4
1 2
2 3
3 4
5 6
6 7
7 8
```
output
```
Yes
6 8 5 7
```

input
```
4
1 2
2 3
3 4
5 6
5 7
5 8
```
output
```
No
```

input
```
7
1 2
1 3
1 4
1 5
5 6
6 7
8 9
9 10
10 11
11 12
12 13
13 14
```
output
```
Yes
9 14 11 12 13 10 8
```

Note

The first sample (bus lines in red and rail lines in blue):