Problem A. Announcements

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

There are N billboards with announcements near Kyoto University.

The i-th billboard appears at day S_i. However, at each T-th day, all billboards installed before this day are removed. You may assume that, on those days, no new billboards will appear.

Find the minimal number of times you need to visit the university to see each billboard at least once.

Input

The first line of input contains one integer N ($1 \le N \le 2 \cdot 10^5$). The second line contains N integers $S_1, S_2, \ldots, S_N$. Here, S_i is the day when the i-th billboard appears ($1 \le S_i \le 10^9$). The last line contains one integer T ($2 \le T \le 10^9$, S_i is not divisible by T for any i): the interval between successive deletions. This means the billboards are removed on days T, $2T$, $3T$, and so on.

Output

Print one integer: the minimum number of visits you need to do to see each billboard at least once.

Examples

standard input
3 1 2 5 3
standard output
2

standard input
5 1 1 1 1 1 2021
standard output
1

standard input
9 623690081 433933447 476190629 262703497 211047202 971407775 628894325 731963982 822804784 128512451
standard output
7

Note

In Example 1, the first two billboards are appearing on days 1 and 2. Then those 2 billboards are removed on day 3. After that, on day 5, the last billboard appears, which is then removed on day 6. So you may visit on day 2 (to see billboards 1 and 2) and on day 5 (to see billboard 3), two times in total.

Problem B. Build The Grid

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Given is a square grid of $N \times N$ squares. Your task is to paint each square of the grid either white or black such that:

- The white squares are connected: for any two white squares, you can go from one to the other by moving only between white squares that share a side.
- Each black square shares a side with at least one white square.
- Denote the number of black cells in the i-th row as p_i. The sequence $P = (p_1, p_2, \ldots, p_N)$ is then a permutation of integers between 0 and $N - 1$, inclusive.
- Denote the number of black cells in the j-th column as q_j. The sequence $Q = (q_1, q_2, \ldots, q_N)$ is then a permutation of integers between 0 and $N - 1$, inclusive.

It can be shown that such a construction always exists.

Input

The input consists of one integer N $(2 \leq N \leq 500)$.

Output

Print N lines. On the i-th line, print a string of length N consisting of characters 'B' and 'W'. The j-th character in the i-th string corresponds to the square in i-th row and j-th column: 'B' denotes black squares and 'W' denotes white squares.

Example

standard input	*standard output*
3	WWB BWB WWW

Problem C. Coins and Boxes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

There are N boxes and N coins on the coordinate line. The coordinate of the i-th box is B_i, and the coordinate of the j-th coin is C_j. You are starting at the point with coordinate 0, and can move freely along the coordinate line.

If you go to a point with a coin, you can pick up that coin. You can carry as many coins as you like. If you go to a point with the box, you can utilize one coin and open the box (but you are not forced to do that). You cannot pick up the coin that was already picked up, or open the box that is already opened.

You want to open all N boxes. Find the minimum distance you need to travel to achieve your goal.

Input

The first line of input contains one integer N ($1 \le N \le 10^5$).

The second line contains N integers $B_1, B_2, \ldots, B_N$. The i-th of those integers is coordinate of the i-th box ($1 \le B_i \le 10^9$, $B_i < B_{i+1}$ for $1 \le i < N$).

The third line contains N integers $C_1, C_2, \ldots, C_N$. The i-th of those integers is coordinate of the i-th coin ($1 \le C_i \le 10^9$, $C_i < C_{i+1}$ for $1 \le i < N$).

Output

Print one integer: the minimum distance you need to travel to open all boxes.

Examples

standard input	*standard output*
4 1 6 7 12 3 5 10 11	21
2 1 2 1 1000000000	1999999998

Problem D. Destructive Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

There are N stone piles, numbered by sequential integers from 1 to N. The i-th pile contains a_i stones. Additionally, each pile i has an integer b_i associated with it.

Alice and Bob play the following game using those stone piles.

They are alternately performing the following operation: choose pile i and a nonnegative integer k such that b_i^k is not greater than the current number of stones in pile i, and remove b_i^k stones from pile i. If a player cannot do that on their turn, the opposite player wins.

Alice moves first. Determine who will win if both players are playing optimally.

Input

The first line of input contains one integer N ($1 \le N \le 10^5$), the number of piles. The i-th of the following N lines contains two integers a_i and b_i ($1 \le a_i, b_i \le 10^9$): the initial number of stones in the i-th pile and the integer associated with it, respectively.

Output

If Alice wins the game when both sides are playing optimally, print "`Alice`". Otherwise, print "`Bob`".

Examples

standard input	standard output
2 10 3 7 4	Bob
16 903 5 246 38 884 12 752 10 200 17 483 6 828 27 473 21 983 35 953 36 363 35 101 3 34 23 199 8 134 2 932 28	Alice
16 35 37 852 17 789 37 848 40 351 27 59 32 271 11 395 20 610 3 631 33 543 14 256 28 48 8 277 24 748 38 109 40	Bob

Problem E. Edges, Colors and MST

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

There is an undirected simple connected graph G with N vertices and M edges. The vertices of G are numbered from 1 to N, and the edges are numbered from 1 to M. Edge i connects vertices u_i and v_i.

Given is a sequence $C = (c_1, c_2, \ldots, c_M)$ of length M, consisting of 0s and 1s. Edge i is painted blue when $c_i = 0$, and is painted red when $c_i = 1$. The edges are colored in such a way that there are exactly $N - 1$ red edges and they are forming a spanning tree of G.

Find the lexicographically smallest permutation $P = (p_1, p_2, \ldots, p_M)$ that satisfies the following condition: if, for each i, the weight of edge i is p_i, then all the edges used in the minimal spanning tree of G are red.

Note that the minimal spanning tree of G is uniquely determined under those conditions.

Input

The first line of input contains two integers N and M: the number of vertices and edges in graph G, respectively ($2 \le N \le 2 \cdot 10^5$, $N - 1 \le M \le 2 \cdot 10^5$).

The following M lines contain descriptions of the edges. Each description contains three integers a_i, b_i and c_i ($1 \le a_i, b_i \le N$, $0 \le c_i \le 1$): the vertices that are connected by this edge and the color of the edge (red if $c_i = 1$ and blue otherwise).

You may assume that there are no multiple edges nor loops, that the given graph is connected, and that the red edges are forming a spanning tree of the given graph.

Output

Print M integers that form the lexicographically smallest permutation P that satisfies the following condition: if, for each i, the weight of edge i is p_i, then all the edges used in the minimal spanning tree of G are red.

Example

standard input	*standard output*
4 5 1 2 0 2 3 1 3 4 1 2 4 0 1 3 1	3 1 4 5 2

Problem F. Flatland Currency

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

The Flatland currency system uses coins of 500, 100, 50, 10, 5, and 1 Flatland yen.

At the shop in the Flatland airport, there are N bottles of milkohol on sale; the i-th bottle costs a_i yen. Note that there are exactly N **bottles**, so you can buy each bottle no more than once.

You have X flatland yen, and you noticed that the number of coins you have is minimal possible between all representations of X.

In the shop, you can do the following sequence of actions any number of times:

- Select some bottles.
- Pay some of the coins you have for the selected bottles.
- The shop returns the change (if needed) using the least possible number of coins. You may assume that the shop will never go short in any type of coins.

You promised your friends 1-yen coins as souvenirs. Find the maximum number of 1-yen coins that you can collect in this shop.

Input

The first line of input contains two integers N and X ($1 \le N \le 10^5$, $1 \le X \le 10^{14}$): the number of bottles in the shop and the number of Flatland yens you have, respectively. The second line contains N integers $A_1, A_2, \ldots, A_N$ ($1 \le A_i \le 10^9$): the prices of the bottles in the shop.

Output

Print one integer: the maximum number of 1-yen coins you may have after visiting the shop.

Examples

standard input	*standard output*
5 57 9 14 31 18 27	8
4 50 11 11 11 11	12

Problem G. Game with Balls and Boxes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

There are N boxes and N balls. You are playing a game that goes as follows.

The boxes are enumerated by sequential integers from 1 to N, and the balls are also enumerated by sequential integers from 1 to N. The i-th box initially contains the ball P_i.

Each box is either open or closed. Initially, all boxes are closed.

Then two rounds of ball movement are performed. In each round, you:

1. Select zero or more boxes and open them. To open the box i for the first round, you pay A_i coins. To open the box i for the second round, you pay B_i coins.
2. Move the balls freely between the open boxes. However, each box must contain exactly one ball when the move is complete.
3. Close all open boxes.

After two rounds, for each i, the box i must contain the ball i. Find the minimal sum of coins you shall pay to complete the game.

Input

The first line of input contains one integer N ($1 \le N \le 10^5$).

The second line contains N integers $P_1, P_2, \ldots, P_N$: here, P_i is the number of the ball that was initially placed in i-th box ($1 \le P_i \le N$, $P_i \ne P_j$ if $i \ne j$).

The third line contains N integers $A_1, A_2, \ldots, A_N$: here, A_i is the price of opening the i-th box for the first round ($1 \le A_i \le 10^9$).

The fourth line contains N integers $B_1, B_2, \ldots, B_N$: here, B_i is the price of opening the i-th box for the second round ($1 \le B_i \le 10^9$).

Output

Print one integer: the minimal sum of coins you need to pay to have i-th ball in the i-th box for each i after two rounds.

Examples

standard input	*standard output*
5 5 3 2 1 4 3 8 3 5 11 9 3 7 6 4	28
1 1 1000000000 1000000000	0

Problem H. High Powers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Given are integers s, t, and u.

Let a, b, and c be distinct complex numbers that satisfy the following conditions:

- $a + b + c = s$,
- $ab + bc + ca = t$,
- $abc = u$.

It is guaranteed that such a, b, and c exist for the given s, t, and u.

Given positive integers n and m, calculate the ratio

$$\frac{a^n(b^m - c^m) + b^n(c^m - a^m) + c^n(a^m - b^m)}{(a-b)(b-c)(c-a)}$$

modulo 998 244 353.

Input

The first line of input contains two integers n and m ($1 \le n, m \le 10^{18}$).

The second line contains three integers s, t and u ($0 \le s, t, u < 998\,244\,353$).

It is guaranteed that the distinct complex numbers a, b, and c from the statement exist for the given s, t, and u.

Output

It can be shown that the answer can be represented as a rational number p/q where p and q are integers, $(p, q) = 1$, $q > 0$ and q is not divisible by 998 244 353.

Print the integer x such that $0 \le x < 998\,244\,353$ and $qx - p$ is divisible by 998 244 353.

Examples

standard input	*standard output*
2 3 314 159 265	159
1000000000000000000 800000000000000000 6 11 6	76083766
1000000000000000000 500000000000000000 505459328 165146837 982639180	228155372

Problem I. Items and Heroes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

There is a rooted tree of N vertices. The vertices are numbered by integers from 1 to N, with vertex 1 as the root. The parent of vertex i $(2 \le i \le N)$ is denoted as P_i.

Each vertex has a box with items. Also, there is a hero in each vertex.

In the beginning, the box in vertex i contains A_i items.

In each vertex i, the hero from that vertex has the quest to collect C_i items. The hero in vertex i can choose some vertices from the subtree rooted at vertex i and take as many items as she wants from each of the selected vertices. One item cannot be taken by more than one hero.

Determine if it is possible for the heroes to act in such a way that all N quests will be completed.

Additionally, Q queries are given. In the j-th query, the integers t_j, v_j, x_j are given, and the values are changed as follows:

- If $t_j = 1$, change the value of A_{v_j} to x_j.
- If $t_j = 2$, change the value of C_{v_j} to x_j.

The queries are applied sequentially. The changes made in each query **remain** for all the subsequent queries as well. After each query, determine if it is possible to complete all N quests.

Input

The first line of input contains one integer N $(1 \le N \le 10^5)$.

The second line contains $N-1$ integers $P_2, P_3, \ldots, P_N$: the parents of vertices $2, 3, \ldots, N$ $(1 \le P_i < i)$.

The third line contains N integers $A_1, A_2, \ldots, A_N$ $(1 \le A_i \le 10^9)$.

The fourth line contains N integers $C_1, C_2, \ldots, C_N$ $(1 \le C_i \le 10^9)$.

The fifth line contains one integer Q $(1 \le Q \le 10^5)$.

Each of the following Q lines contains one query described by three integers t_j, v_j and x_j $(1 \le t_i \le 2$, $1 \le v_i \le N$, $1 \le x_i \le 10^9)$: the type of the query, the number of vertex and the new value for A_{v_i} (for the query of the first type) or C_{v_i} (for the query of the second type), respectively.

Output

On the first line, print "`Yes`" if it is possible to complete all N quests at once, or "`No`" otherwise.

On the following Q lines, print the answers for the queries in the same format, one per line.

Examples

standard input
3 1 1 2 1 3 3 1 2 2 1 1 1 2 3 1
standard output
Yes No Yes

standard input
5 1 2 1 3 1000000000 1000000000 1000000000 1000000000 1000000000 1 1 1 1 1 1 1 1 1
standard output
Yes Yes

standard input
5 1 2 2 2 109102235 645590056 708566822 497603443 131863700 50073184 441114664 164994352 304489019 158100373 8 1 5 692234112 1 3 610338520 2 4 818442884 2 4 164762830 2 4 923652447 2 4 197720766 1 1 779302743 1 1 222486377
standard output
No Yes Yes No Yes No Yes Yes Yes

Note

In Example 1, the hero from vertex 1 takes two items from the box at vertex 1 and one item from the box at vertex 3, the hero from vertex 2 takes an item from the box at vertex 2, and the hero from vertex 3 takes two items from the box at vertex 3. So, all three quests are completed.

The first query changes the number of items in the box at vertex 1 from two to one. In this case, there are not enough items to complete all three quests.

The second query changes the number of items to complete the quest for the hero at vertex 3 from two to one. In this case, the hero at vertex 1 takes one item from the box at vertex 1 and two items from the box at vertex 3, the hero at vertex 2 take one item from the box at vertex 2, the hero at vertex 3 takes one item from the box at vertex 3, and all three quests are again completed.

Problem J. Juggler's Trick

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 megabytes

N balls are lined up in a row from left to right. Each ball may be either uncolored (white), blue, or red. Additionally, two integers r and b are given. Let us represent the ordering as a string consisting of letters 'W', 'B' and 'R' for uncolored, black, and red balls, respectively.

For each trick, the juggler may choose a *combo* of $r + b$ consecutive balls such that there are exactly r red balls and exactly b blue balls, in any order, and remove them. The remaining balls are concatenated while keeping their relative order. For example, if the initial order was "RRBRBBR", and the juggler removed "RBB", the result would be "RRBR".

Before the process starts, the juggler shall paint each uncolored ball either red or blue. The juggler wants to do as many tricks as possible. Find the maximal number of tricks if the juggler will choose the colors for the uncolored balls optimally.

Input

The first line of input contains three integers N, r and b ($1 \le N \le 2 \cdot 10^5$, $1 \le r, b \le N - 1$, $r + b \le N$): the total number of balls, the number of red balls in a combo and the number of the blue balls in a combo, respectively. The second line contains the string S. This string encodes the initial order of balls and consists of exactly N letters 'B', 'R' and 'W', representing blue, red, and uncolored balls, respectively.

Output

Print one integer: the maximum number of tricks that can be done by the juggler.

Examples

standard input	*standard output*
4 1 1 BBWR	2
6 2 1 RBBBWB	0
13 3 3 WWWWWWWWWWWWW	2

Note

In Example 1, the juggler paints the white ball in red, obtaining the order "BBRR", then removes combo "BR"; the remaining balls have order "BR", so they can be removed. Since there are 4 balls initially, and after each trick, exactly two balls are removed, 2 is the maximal possible number of tricks that can be done.

In Example 2, the juggler cannot obtain any sequence of 3 balls with two red and one blue ball regardless of the coloring of the white ball, so the answer is 0.

Problem K. King's Palace

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	1024 mebibytes

There are N walls in the hall of the King's palace, numbered by integers from 1 to N. The King asks the Royal Painter to paint each wall in one of three colors (red, green, or blue). Additionally, the King gives M orders.

Every order has the following form: given two walls, a_i and b_i, and two colors, x_i and y_i, the order dictates that, if the wall a_i is painted with color x_i **and** the wall b_i is painted with color y_i, the Royal Painter has to be executed.

Your task is to find a number of ways to paint the walls so that the Royal Painter will not be executed.

Input

The first line of the input contains two integers N and M ($1 \le N \le 22$, $1 \le M \le 9 \cdot N \cdot (N-1)/2$): the number of walls and the number of orders, respectively.

Each of the following M lines describes one King's order and contains an integer a_i, a letter x_i, an integer b_i, and a letter y_i, separated by single spaces ($1 \le a_i < b_i \le N$, x_i and y_i are letters from 'R', 'G', and 'B', denoting the red, green, and blue colors, respectively). You may assume that all M orders are pairwise distinct (no two orders have the exact same effect).

Output

Print one integer: the number of ways to paint the walls so that the Royal Painter will not be executed.

Examples

standard input	*standard output*
2 3 1 R 2 R 1 G 2 R 1 B 2 G	6
1 0	3
22 0	31381059609
4 12 2 R 3 R 1 B 2 B 2 R 3 B 3 R 4 R 1 B 4 G 1 R 3 B 3 G 4 B 2 G 3 G 1 B 2 R 1 G 2 R 1 R 3 G 1 G 3 B	13

Problem L. Lion and Zebra

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Given is a tree with N vertices.

There is a tag game played on this tree. The game consists of several rounds.

In each round, there are two players: the *lion*, which is the catching side, and the *zebra*, the escaping side.

At the beginning of each round, the zebra and the lion start in two distinct vertices. The lion always knows where the zebra is, and chases it at a speed of one edge per second. The zebra does not know the position of the lion, but always knows the distance to the lion. Based on that information, the zebra makes one of the following two choices each second:

- Take 1 second to move to any adjacent vertex.
- Stay at the current vertex for 1 second.

When the zebra meets the lion on an edge or in a vertex, the round is over. If the players move by the same edge towards each other, the meeting happens 0.5 seconds after they start moving. Zebra acts in a way that the minimum (of all possible initial vertices for the lion) time to meet the lion is maximized.

You are given Q rounds. In the i-th round, zebra starts from the vertex v_i, and the distance to the lion is equal to d_i. For each round, find the minimum time when this round will be over if both sides will follow their strategies.

Input

The first line of input contains two integers N and Q: the number of vertices in the tree and the number of rounds played ($2 \le N \le 10^5$, $1 \le Q \le 10^5$).

Each of the following $N - 1$ lines contains two integers a_i and b_i: the vertices that are connected by an edge. You may assume that the given graph is a tree.

Each of the following Q lines describes one round and contains two integers v_j and d_j ($1 \le v_j \le N$, $1 \le d_j \le N - 1$): the starting vertex for the zebra and the distance from the zebra to the lion at the beginning of this round. You may assume that there exists at least one vertex w_j such that the distance between v_j and w_j is equal to d_j.

Output

For each query, print one integer: the minimum time of round when both sides follow their strategies.

Examples

standard input	standard output
5 2 1 2 2 3 3 4 4 5 1 4 3 1	4 1
11 2 1 2 2 3 1 4 4 5 1 6 6 7 7 8 1 9 9 10 10 11 3 2 10 4	2 5

Note

At the beginning of the first round in Example 1, the zebra is at vertex 1, and the distance to the lion is 4, so we can conclude that the lion is at vertex 5. In this case, the optimal strategy is to stay at vertex 1 as long as possible, and the answer is equal to 4.

At the beginning of the second round, the zebra is at vertex 3, and the distance to the lion is 1, so we can conclude that the lion is either at vertex 2 or at vertex 4.

If the zebra moves in the direction of vertex 2, the lion will meet the zebra on the edge after 0.5 seconds if it started at vertex 2, or at vertex 1 after 3 seconds if it started at vertex 4. Therefore, the minimum possible time until the zebra meets the lion with this strategy is 0.5.

Similarly, if the zebra moves in the direction of vertex 4, the minimum possible time until encountering the lion is also 0.5.

If the zebra stays at vertex 3, they would meet after 1 second if the lion started at vertex 2, or after 1 second if the lion started at vertex 4. Therefore, the answer is equal to 1.

Problem M. Math String

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 1024 mebibytes

Consider a string S of length N composed of 11 characters: '1', '2', '3', '4', '5', '6', '7', '8', '9', '+', '*'.

We will call string S a *math string* if:

- The first and last characters of S are neither '+' nor '*'.
- When two consecutive characters are taken from S, at least one of them is neither '+' nor '*'.

Each math string can be treated as an arithmetic expression with integers in decimal notation using ordinary arithmetic operations, where multiplication takes precedence over addition. For each such expression, its value can be calculated: for example, the value of math string "35+2*6" is 47. Please find the sum of the values of all math strings of the given length N, modulo 998 244 353.

Input

The input contains one integer N ($1 \le N \le 10^{18}$).

Output

Print one integer: the answer to the problem.

Examples

standard input	standard output
1	45
3	407430
1000000000000000000	493565653

Note

In the Example 1, there are only 9 distinct one-digit math strings: the digits from '1' to '9'. The sum of those digits, treated as arithmetic expressions, is equal to 45.