

Problem A

Dyad

Time Limit: 2 seconds

A shop sells sweets in pairs.

There are N products. The i -th product contains a pair of sweets: one of type A_i and one of type B_i . A_i and B_i may be equal.

If you buy two different products, you will obtain four sweets in total. There are twins, and you want to give exactly two sweets to each of them. The twins are only satisfied if the multiset of sweet types received by each twin is identical.

Count the number of ways to choose two different products such that it is possible to distribute the sweets to the twins in this way.

Input

The input is given in the following format:

N
 $A_1 B_1$
 $A_2 B_2$
 $\vdots$
 $A_N B_N$

- $2 \leq N \leq 300\,000$
- $1 \leq A_i, B_i \leq N$ ($1 \leq i \leq N$)
- All input values are integers.

Output

Output the answer in a single line.

Sample Input	Sample Output
4 2 3 3 2 1 1 2 2	2

Problem B

JAG Box

Time Limit: 2 seconds

The JAG Box is an ordinary rectangular box that is currently popular around the world. There are N JAG Boxes. For each $i = 1, 2, \dots, N$, the i -th box has an integer weight A_i .

You will build a vertical stack by repeatedly choosing one remaining box and inserting it at the very bottom of the current stack. When a box of weight w is inserted at the bottom of the existing stack whose total weight is x , that box receives load equal to $\left\lfloor \frac{x}{w} \right\rfloor$.

Determine the minimum possible total load over all boxes.

Input

The input is given in the following format:

N
 $A_1 A_2 \dots A_N$

- $2 \leq N \leq 200\,000$
- $1 \leq A_i \leq 10^9$ ($1 \leq i \leq N$)
- All input values are integers.

Output

Output the answer in a single line.

Sample Input	Sample Output
5 3 1 4 1 5	3

Problem C

Chairs

Time Limit: 2 seconds

There are HW chairs arranged in H rows and W columns. We denote the chair in the i -th row from the top and the j -th column from the left by (i, j) .

Some chairs may have luggage placed on them. The situation of the chairs is represented by H strings $S_1, S_2, \dots, S_H$, each of length W . If the j -th character of S_i is '#', then there is luggage on (i, j) . If it is '.', then there is no luggage on (i, j) . It is guaranteed that there is at least one chair on which there is no luggage.

We want to seat people on these chairs. At most one person can sit on each chair, and a person cannot sit on a chair that has luggage on it. Moreover, two people cannot sit on chairs that are adjacent to each other vertically or horizontally. Under these conditions, we want to seat as many people as possible. Let M be the maximum number of people we can seat observing these rules.

Now, suppose one person arrives. For each chair, determine whether we may seat this person there. Specifically, determine whether it is possible to seat this person on that chair, and in addition, still be able to seat $M - 1$ more people under the rules.

Input

The input is given in the following format:

```
 $H$   $W$ 
 $S_1$ 
 $S_2$ 
 $\vdots$ 
 $S_H$ 
```

- $1 \leq H \leq 400$
- $1 \leq W \leq 400$
- S_i is a string of length W consisting of '#' and '.' ($1 \leq i \leq H$).
- There exists (i, j) such that the j -th character of S_i is '.'.
- H and W are integers.

Output

Output H lines. On the i -th line ($1 \leq i \leq H$), output a string of length W .

For each (i, j) , if we can seat the newly arrived person on (i, j) , then the j -th character of the string on the i -th line must be '1'. Otherwise, it must be '0'.

Sample Input	Sample Output
3 4 ##.. #.##	0011 1011 0100

Problem D

Inversion of Suffix Array

Time Limit: 2 seconds

You are given positive integers N, K and a string S of length N consisting of lowercase English letters.

Let T be the string obtained by concatenating K copies of S .

Find the inversion number of the Suffix Array of T , modulo 998244353.

For a string s of length n , the suffix array of s is a permutation of integers from 1 to n that represents the starting positions of all non-empty suffixes of s , sorted in lexicographical order.

Input

The input is given in the following format:

N K
 S

- $1 \leq N \leq 200\,000$
- $1 \leq K \leq 10^{12}$
- S is a string of length N consisting of lowercase English letters.
- N and K are integers.

Output

Output the answer in a single line.

Sample Input 1	Sample Output 1
4 2 icpc	18

Sample Input 2	Sample Output 2
13 92025 jagsummercamp	996562345

Problem E

Pole

Time Limit: 2 seconds

On the sphere centered at $(0, 0, 0)$ with radius R , there are N circles. The i -th circle is defined as the set of intersection points between the sphere and the following plane:

- The plane passes through the point (X_i, Y_i, Z_i)
- The plane is orthogonal to $\vec{v} = (X_i, Y_i, Z_i)$.

It is guaranteed that $v \neq \vec{0}$.

It is also guaranteed that no two circles share any common point.

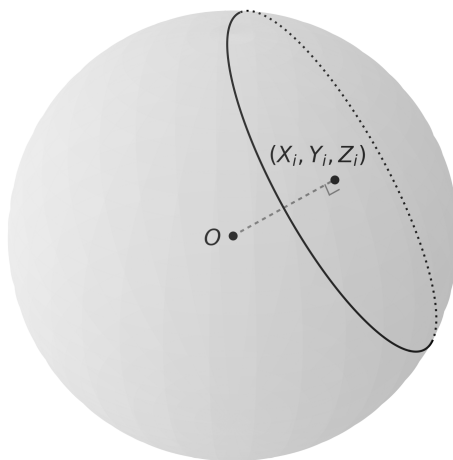


Figure E-1: The circle defined by (X_i, Y_i, Z_i)

We define the *distance* between two points on the sphere as follows:

For a path on the sphere connecting these two points, count how many of the circles the path intersects. The distance is the minimum possible value of this count over all such paths.

You may choose one point on the sphere and designate it as the **Pole**. Find the minimum possible value of

$$\max_{p \in \text{sphere}} \text{distance}(\text{Pole}, p).$$

Input

The input is given in the following format:

```

N R
X1 Y1 Z1
X2 Y2 Z2
⋮
XN YN ZN

```

- $1 \leq N \leq 2000$
- $1 \leq R \leq 10^6$
- $0 < \|(X_i, Y_i, Z_i)\| < R$ ($1 \leq i \leq N$)
- No two circles share any common point.
- All input values are integers.

Output

Output the answer in a single line.

Sample Input	Sample Output
5 100 14 -11 -2 -54 46 8 54 -57 -12 -34 39 7 64 -74 -4	3

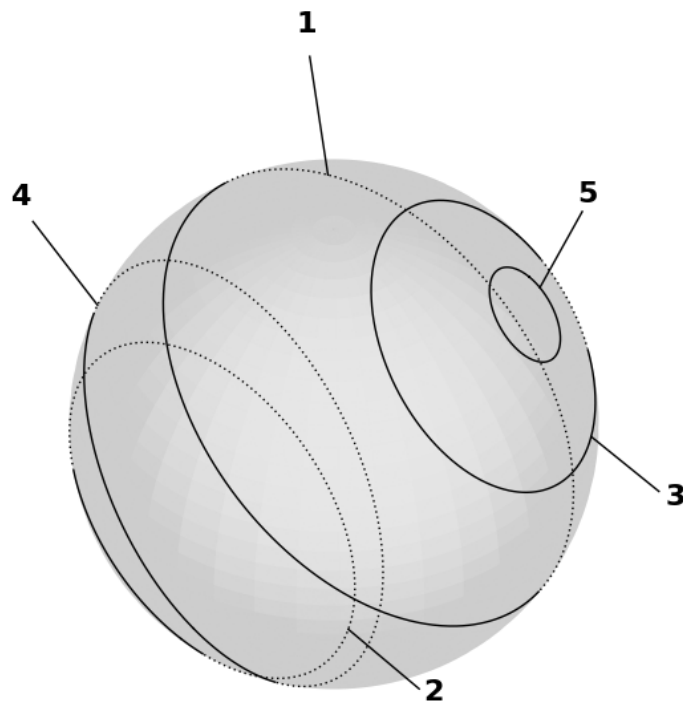


Figure E-2: Illustration of Sample Input

Problem F

Walking on Binary Tree

Time Limit: 2 seconds

You are given an infinite complete binary tree whose vertices are labeled with positive integers. The root is vertex 1, and for every vertex x ($x \geq 2$), its parent is $\left\lfloor \frac{x}{2} \right\rfloor$.

You are also given a string $S = S_0 S_1 \dots S_{N-1}$ of length N . Each character of S is either 'L' or 'R'.

Consider the following process: you are currently at some vertex u , and want to reach vertex v by repeatedly moving through the tree.

On the i -th move (1-indexed), suppose you are at vertex x . You may choose one of the following moves:

Downward move: If $S_{(i-1) \bmod N}$ is 'L', move to vertex $2x$. Otherwise, move to vertex $2x + 1$.

Upward move: You can choose only if $x \geq 2$. Move to vertex $\left\lfloor \frac{x}{2} \right\rfloor$.

Note that you cannot stay at the same vertex.

You are given Q independent queries. In the i -th query, you start at vertex u_i and want to reach vertex v_i .

For each query, determine whether it is possible to reach v_i from u_i . If it is possible, find the minimum number of moves required.

Input

The input is given in the following format:

```
N
S
Q
u1 v1
u2 v2
⋮
uQ vQ
```

- $1 \leq N \leq 10^6$
- $1 \leq Q \leq 200\,000$
- $1 \leq u_i, v_i \leq 10^{18}$ ($1 \leq i \leq Q$)
- N, Q, u_i and v_i are integers.
- S is a string of length N consisting of 'L' and 'R'.

Output

Output Q lines. On the i -th line, print the minimum number of moves required to reach v_i from u_i if it is possible; otherwise, print “-1”.

Sample Input 1

```
5
LLRLR
3
1 12
9 2
913 2025
```

Sample Output 1

```
7
2
23
```

Sample Input 2

```
1
L
1
1 3
```

Sample Output 2

```
-1
```

Problem G

Path Flipping

Time Limit: 2 seconds

For a grid with each cell colored either white or black, let us define the **beauty** of the grid as follows:

- Consider performing the following operation any number of times:
 - Choose a path from the upper-left corner to the lower-right corner, consisting only of downward and rightward moves. Invert the colors of all cells on the chosen path.
- The **beauty** of the grid is defined as the maximum possible number of cells colored black.

You have a grid with H rows and W columns. Initially, all cells are colored white.

You need to process Q queries in order. The i -th query is given in the following format:

- You are given two integers t_i and x_i .
 - If $t_i = 1$, invert the colors of all cells in the x_i -th row from the top.
 - If $t_i = 2$, invert the colors of all cells in the x_i -th column from the left.
- Then, find the **beauty** of the current grid.

Input

The input is given in the following format:

```

H W Q
t1 x1
⋮
tQ xQ

```

- $1 \leq H, W \leq 200\,000$
- $1 \leq Q \leq 200\,000$
- $t_i \in \{1, 2\}$ ($1 \leq i \leq Q$)
- $t_i = 1 \implies 1 \leq x_i \leq H$ ($1 \leq i \leq Q$)
- $t_i = 2 \implies 1 \leq x_i \leq W$ ($1 \leq i \leq Q$)
- All input values are integers.

Output

Output Q lines. On the i -th line ($1 \leq i \leq Q$), output the answer for the i -th query.

Sample Input	Sample Output
3 4 5	9
2 2	12
2 3	10
1 1	10
1 2	9
2 3	

Problem H

Colored Tree and Path

Time Limit: 4 seconds

You are given a tree with N vertices, numbered 1 through N . Edge i connects vertices a_i and b_i . Each vertex i is assigned a color c_i .

You are asked to process Q queries. In each query, four integers u_1, v_1, u_2, v_2 are given.

For each query, determine the maximum integer K ($0 \leq K \leq N$) such that the following condition holds:

- For every $j = 1, 2, \dots, K$, the number of vertices of color j on the path from u_1 to v_1 is equal to the number of vertices of color j on the path from u_2 to v_2 .

Input

The input is given in the following format:

```
N
a1 b1
a2 b2
⋮
aN-1 bN-1
c1 c2 ... cN
Q
Query1
Query2
⋮
QueryQ
```

Each Query is given in the following format:

```
u1 v1 u2 v2
```

- $1 \leq N \leq 100\,000$
- $1 \leq a_i, b_i \leq N$ ($1 \leq i \leq N-1$)
- $1 \leq c_i \leq N$ ($1 \leq i \leq N$)
- $1 \leq Q \leq 100\,000$
- $1 \leq u_1, v_1, u_2, v_2 \leq N$ ($1 \leq i \leq Q$)
- The given graph is a tree.
- All input values are integers.

Output

Output Q lines. On the i -th line ($1 \leq i \leq Q$), output the answer for the i -th query.

Sample Input	Sample Output
6 2 3 4 3 6 2 3 5 2 1 1 2 2 3 1 1 4 1 6 5 4 6 5 1 5 1 1 6 6 1 5 4 2	0 6 6 0

Problem I

Inside Yamanote

Time Limit: 2 seconds

There are N cities numbered from 0 to $N - 1$. A railway goes around all N cities, and city i and city $(i + 1) \bmod N$ can be traveled between in L_i time units.

You will implement the following policy:

- You may construct any number of railways that allow travel between any two cities in any non-negative time.
- You then select one city among the N cities to be the capital. The minimum travel time from the capital to city i using the railways is defined as that city's undevelopment index d_i .

The reputation of this policy will be determined by word of mouth from the M residents who will move this year. Resident j will move from city X_j to city Y_j after the policy is implemented. The reputation will be the sum of $d_{X_j} - d_{Y_j}$.

Your goal is to maximize the reputation of the policy. However, renovation work on the existing railway is also in progress, and you must revise the policy accordingly.

The travel times of existing railways will change Q times. At the k -th change, the travel time between city T_k and city $(T_k + 1) \bmod N$ changes to Z_k . These changes are persistent. After each change, output the maximum possible reputation of the policy under the new conditions.

Input

The input is given in the following format:

```
N M Q
L0 L1 ... LN-1
X1 Y1
X2 Y2
⋮
XM YM
T1 Z1
T2 Z2
⋮
TQ ZQ
```

- $3 \leq N \leq 200\,000$
- $1 \leq M \leq 200\,000$
- $1 \leq Q \leq 200\,000$
- $0 \leq L_i \leq 10^6$ ($1 \leq i \leq N$)
- $0 \leq X_j, Y_j \leq N - 1$ ($1 \leq j \leq M$)
- $X_j \neq Y_j$ ($1 \leq j \leq M$)
- $0 \leq T_k \leq N - 1$ ($1 \leq k \leq Q$)
- $0 \leq Z_k \leq 10^6$ ($1 \leq k \leq Q$)
- All input values are integers.

Output

Output Q lines. On the k -th line ($1 \leq k \leq Q$), output the answer for the query k . It can be proven that the answer is an integer.

Sample Input	Sample Output
5 3 4	8
1 5 2 1 3	7
0 2	8
1 3	15
4 2	
4 0	
1 3	
4 2	
3 9	

Problem J Billion Tree

Time Limit: 2 seconds

This is an interactive problem.

The interaction consists of two phases.

Phase 1

You must first choose an integer N ($2 \leq N \leq 65$). Then you must output the chosen N and a tree T on N vertices satisfying the following conditions:

- The vertices are numbered $1, 2, \dots, N$.
- Each edge of T has an integer weight between 0 and 10^9 , inclusive.

Phase 2

You are given an integer Q , followed by Q integers $x_1, x_2, \dots, x_Q$. For each i ($1 \leq i \leq Q$), you must express x_i as the sum of the weights of at most 5 paths in T .

Formally, for each i you must output:

- an integer K ($1 \leq K \leq 5$),
- and K pairs of vertices $(u_1, v_1), (u_2, v_2), \dots, (u_K, v_K)$,

such that

$$\sum_{j=1}^K w(u_j, v_j) = x_i,$$

where $w(u, v)$ denotes the weight of the path between vertices u and v in T .

The weight of a path is defined as the sum of the weights of the edges contained in the path.

Interaction

The interaction proceeds as follows:

Phase 1

Choose an integer N and a weighted tree T , and output them in the following format:

```

N
a1 b1 c1
⋮
aN-1 bN-1 cN-1
```

Each triple (a_i, b_i, c_i) represents an edge between vertices a_i and b_i with weight c_i . The following conditions must hold:

- $2 \leq N \leq 65$
- $1 \leq a_i, b_i \leq N \quad (1 \leq i \leq N - 1)$
- $0 \leq c_i \leq 10^9 \quad (1 \leq i \leq N - 1)$

Phase 2

First, an integer Q ($1 \leq Q \leq 10\,000$) is given. Then, Q integers $x_1, x_2, \dots, x_Q$ ($1 \leq x_i \leq 10^9$) are given one by one. For each integer x_i , you must output your answer in the following format:

```

K
u1 v1
⋮
uK vK

```

Here K ($1 \leq K \leq 5$) is the number of paths, and each pair (u_j, v_j) represents a path between vertices u_j and v_j .

Note that the value x_i ($i \geq 2$) is provided only after the answer for x_{i-1} has been printed.

Notes on Interactive judging

The verdict will be indeterminate if there is malformed output during the interaction or your program quits prematurely. Terminate the program immediately after printing the answer, or the verdict will be indeterminate. As some environments require flushing the output buffers, make sure that your outputs are actually sent. Otherwise, your outputs will never reach the judge.

Sample Interaction

Read	Write
3	3
110	1 2 10
	3 1 100
	1
210	2 3
	3
	2 1
	1 3
	1 3
20	2
	1 2
	1 2

Problem K

LOGCFL

Time Limit: 2 seconds

You are given a 3-dimensional integer array A of size $N \times N \times N$.

Initialize the following variables:

```
1 integer x = 0;  
2 integer w = 1;  
3 stack s = {};
```

Then, for each $t = 0, 1, \dots, N - 1$, choose an integer y_t such that $-1 \leq y_t < N$ and do the following action:

If $0 \leq y_t$, do the following:

```
1 w *= A[t][x][y];  
2 s.push(x);  
3 x = y;
```

The above y denotes y_t .

If $y_t = -1$, do the following:

```
1 assert(!s.empty());  
2 w *= A[t][x][s.top()];  
3 x = (x + s.top()) % N;  
4 s.pop();
```

You can't choose $y_t = -1$ if the stack is empty before the action.

Note that the stack has the following operations:

- `push(x)`: adds an element x to the collection.
- `pop()`: removes the most recently added element.
- `top()`: returns the value of most recently added element.

For each $i = 0, 1, \dots, N - 1$, consider all possible sequences $y_0, y_1, \dots, y_{N-1}$ such that the final value of x is i . Compute the sum of the corresponding values of w over all such sequences, and output the result modulo 998244353.

Input

The input is given in the following format:

$$\begin{array}{c}
 N \\
 A_{0,0,0} \dots A_{0,0,N-1} \\
 \vdots \\
 A_{0,N-1,0} \dots A_{0,N-1,N-1} \\
 \vdots \\
 \vdots \\
 A_{N-1,0,0} \dots A_{N-1,0,N-1} \\
 \vdots \\
 A_{N-1,N-1,0} \dots A_{N-1,N-1,N-1}
 \end{array}$$

$A_{i,j,k}$ means the value of $A[i][j][k]$.

- $1 \leq N \leq 30$
- $0 \leq A_{i,j,k} \leq 10^9$ ($1 \leq i, j, k \leq N$)
- All input values are integers.

Output

Output N lines. On the i -th line ($0 \leq i < N$), output the answer for i .

Sample Input 1	Sample Output 1
2 1 10 100 1000 1 3 9 27	92 363

Sample Input 2	Sample Output 2
3 2 1 2 1 2 1 1 1 1 2 1 1 2 2 1 2 2 2 2 1 1 1 2 1 1 2 2	63 68 56

Sample Input 3

```
4
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
```

Sample Output 3

```
120
120
120
120
```

Problem L

Sum of Floor(N/ij)

Time Limit: 4 seconds

You are given positive integer N . Find the value

$$\sum_{i=1}^N \sum_{j=1}^N \left\lfloor \frac{N}{ij} \right\rfloor.$$

You are given T test cases, so find the answer for each.

Input

The input is given in the following format:

```
 $T$ 
case1
case2
⋮
case $T$ 
```

case _{i} represents the i -th test case. Each test case is given in the following format:

N

- $1 \leq T \leq 100$
- $1 \leq N \leq 10^9$
- All input values are integers.

Output

Output T lines. On the i -th line ($1 \leq i \leq T$), output the answer to the i -th test case.

Sample Input	Sample Output
10	1
1	53
10	1471
100	29425
1000	496623
10000	7518850
100000	106030594
1000000	1421760251
10000000	18362473634
100000000	230375375227
1000000000	