Problem A. Accommodation Plan

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

The big party will be held at Zenyk and Marichka's home. Their K friends will arrive to them.

Now Zenyk tries to find a way to accommodate all friends. Their house consists of N rooms and $N-1$ hallways. Each hallway connects 2 rooms and has some length. It's possible to reach any room starting from any room via hallways.

Zenyk calls accommodation plan good if

- Each friend lives in some room.
- No 2 friends live in the same room.
- There exist a room (doesn't matter if someone lives there) such that all friends can meet in this room and the distance from it to room of each friend is not bigger than L.

Now Zenyk wants to count the number of good accommodation plans. Two plans are considered different if at least one friend lives in different rooms. As this number can be very big, print it modulo 1000000007.

Input

The first line contains 3 integers – N, K and L ($1 \le K \le N \le 10^5$, $1 \le L \le 10^9$). Each of the next $N-1$ lines contain 3 integers – A_i, B_i and C_i ($1 \le A_i, B_i \le N$, $1 \le C_i \le 10^9$.), which mean that a hallway connecting A_i and B_i exists with length C_i.

Output

Print one integer – number of good accommodation plans modulo 1000000007.

Example

standard input	standard output
5 2 7 1 2 4 3 2 8 2 4 2 4 5 6	12

Note

All good accommodation plans:

$\{1,2\},\{1,4\},\{1,5\},\{2,1\},\{2,4\},\{2,5\},\{4,1\},\{4,2\},\{4,5\},\{5,1\},\{5,2\},\{5,4\}$.

A pair of integers represents room indices of the first and the second friend, respectively.

Problem B. Card Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Zenyk and Marichka start playing Marichka's favourite card game. But Zenyk forgets even how the deck looks like. Of course, he can't just ask Marichka.

He knows that deck contains N cards. Also there are some number of suits, let's say it's K. Number of cards of each suit is the same. Suits are numbered from 1 to K.

At the beginning of the game M cards were dealt. So Zenyk knows suits of these cards. Help Zenyk to find if it's possible to determine the value of K uniquely. Note that the deck is valid so at least one valid value of K exists.

Input

The first line contains 2 integers – N and M ($1 \le N \le 10^9$, $1 \le M \le \min(N, 10^5)$).

The second line contains M integers A_i – suit of the i-th card ($1 \le A_i \le N$).

Output

Print "`YES`" if it's possible to determine number of suits uniquely and "`NO`" otherwise.

Examples

standard input	standard output
36 11 1 4 2 4 4 2 4 1 4 4 4	YES
4 2 1 1	NO

Note

In the first test the only valid situation is 4 suits 9 cards each.

In the second test there can be 1 or 2 suits.

Problem C. The Most Expensive Gift

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Zenyk wants to make the best gift in the world for Marichka. He is visiting a gift shop and wants to buy the most expensive one.

But this is not a typical store, they have only string S consisting of letters 'a', 'b' and 'c'. Zenyk can choose any subsequence of this string as a gift. Subsequence T is a string that can be derived from S by deleting some (possible none) letters without changing the order of the remaining letters. Price of subsequence T equals $\frac{len_T^2}{c_T}$, where len_T is the length of subsequence T and c_T is the length of the smallest cycle of T.

String R is the cycle of string T if

- $length(R)$ is a divisor of $length(T)$.
- $R_{i\%length(R)} = T_i$ for all $i \in [0, length(T) - 1]$ (indexing from 0).

Help Zenyk to find the most expensive gift.

Input

The first line of the input contains one integer N, which denotes the length of the string S ($1 \leq N \leq 10^4$). Second line contains string S consisting of letters 'a', 'b' and 'c'.

Output

Print one integer – the price of the most expensive gift.

Example

standard input	standard output
11 abcabacbcac	18

Note

One of the most expensive subsequences is "ababab". It's length equals 6 and length of the smallest cycle equals 2. So price of this subsequence equals $\frac{6^2}{2} = 18$. There are another subsequences with the same price.

Problem D. Cut the Cake

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

It is Marichka's k^2-th birthday today! Zenyk bought a big cake for this occasion and now he wants to cut it.

For the sake of simplicity, consider the cake as rectangular matrix with n rows and m columns. There are exactly k^2 candles on it, each of them located in a unique cell of the matrix. Zenyk wants to cut the cake with $k-1$ horizontal and $k-1$ vertical cuts. (Note that he's only allowed to cut between cells.) After the cutting, each of k^2 parts must contain a single candle.

You task is to find and output any valid cutting, or indicate that it's impossible to achive the goal.

Input

The first line contains three integers n, m and k ($2 \le k \le n, m \le 200$). The following n lines contain a string of m characters each. Character '1' represents a cell with a candle on it, while '0' respresents a cell without candle.

It's guaranteed that there are exactly k^2 candles on the cake.

Output

In the first line print "YES" if it's possible to cut the cake the way Zenyk wants, otherwise print "NO".

In case of positive answer the second line must contain $k-1$ unique valid indices of the horizontal cuts, and the third line must contain $k-1$ unique valid indices of the vertical cuts. A cut between rows (or columns) i and $i+1$ has index i (1-based).

Examples

standard input	standard output
4 4 2 1000 0001 0010 0001	YES 2 3
3 4 2 1110 0000 0100	NO

Problem E. Message

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

One evening Zenyk decided to send some nice message to Marichka to congratulate her on the start of spring. Initially, he entered message s on his phone, but after a moment he realized, that it would be much better to enter message t.

Unfortunately, it's not so easy to change the message now – the only thing Zenyk is able to do is to remove the first or the last occurrence of any letter. Please note that he is able to perform this operation any number of times. Moreover, the letters are not removed instantly. It takes w_i seconds to remove character that was initially placed on i-th position in string s.

Help Zenyk to calculate the minumum number of seconds it takes to transform message s into t using the described operations. If it's impossible to do that, print a single line "`You better start from scratch man...`" (without quotes).

Input

The first line of the input contains string s, the second – string t ($1 \le |s|, |t| \le 200000$). Strings s and t consist only of the lower case latin letters `a-z`.

The third line contains $|s|$ space-separated integers w_i, each of which denotes the number of seconds it takes to remove the corresponding character ($1 \le w_i \le 10^9$).

Output

If Zenyk is able to transform s into t, print the minimum number of seconds required to do that. Otherwise, print "`You better start from scratch man...`" (without quotes).

Examples

standard input	standard output
ababccb abc 7 2 2 4 3 2 1	7
babab baab 2 1 3 2 4	You better start from scratch man...

Problem F. Bad Word

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Marichka has seen bad word while surfing the net! She immediately starts crying and asks Zenyk to destroy it.

Zenyk knows that Marichka saw word S which consists of lower case english characters. Zenyk can delete any substring $S_l S_{l+1} S_{l+2} \ldots S_r$ of this word in one minute. But he knows that Marichka is keen on palindromes so if this substring is palindrome, Marichka will resent. Zenyk decided that he wouldn't delete such substrings. Now Zenyk wants to know minimum time to destroy bad word or if it is impossible.

Palindrome is such string that reads the same backward as forward. For example, strings "`bob`", "`abba`", "`aaaa`" are palindromes and "`cat`", "`dog`", "`penguin`" are not.

Input

First line contains one integer N – length of the word S ($1 \leq N \leq 10^5$). Second line contains word S which consists of lower case english characters.

Output

Print minimum number of minutes to destroy bad word or -1 if it is impossible.

Examples

standard input	standard output
7 abcdcba	2
3 xxx	-1

Note

In the first case Zenyk can delete substring "`bcd`" during the first minute. Remaining word equals "`acba`" and can be deleted during the second minute.

Problem G. Zenyk, Marichka and Interesting Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

As you may know Zenyk and Marichka are together for years. They will live forever in our hearts. And problem statements.

But they are bored of such casual life. So they decide to play an interesting game. Zenyk and Marichka have N piles of stones. i-th pile contains X_i stones. Zenyk and Marichka take turns alternately. Zenyk moves first. Zenyk takes exactly A stones from any pile in one move (of course, this pile should contain at least A stones). Marichka takes exactly B stones from any pile.

Player who cannot make a move, that means that any pile contains enough stones, loses the game.

Marichka is interested who will win if both players play optimally.

Input

The first line contains 3 integers N, A, B. ($1 \leq N \leq 10^5$, $1 \leq A, B \leq 10^9$).

The second line contains N integers X_i ($1 \leq X_i \leq 10^9$).

Output

Print "`Zenyk`", if Zenyk will win and "`Marichka`", otherwise.

Example

standard input	standard output
`4 4 7` `7 2 14 7`	`Marichka`

Problem H. Frog Jumping

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

Marichka and Zenyk like romantic evenings. Today they decided to watch frogs jump over stones.

There are n stones placed in a line and numbered from the left to the right using integers from 1 to n, inclusive. The distance between any two consecutive stones is exactly 1 meter.

There are also m frogs, initially located on the first stone. The objective is to move all frogs to the last (n-th) stone by jumping. Each frog can only jump forward.

The following two condition must be fulfilled:

1. Stones a_1, a_2, ..., a_k must be visited by exactly one of the frogs.
2. All the other stones (except the first one and the last one) must be never visited by any frog.

When the i-th frog jumps more than d meters in a single jump, it costs c_i units of energy. Any smaller jump costs nothing.

Your task is to find the minimum total amount of energy needed for all frogs to get to the last stone.

Input

The first line of the input contains four space-separated integers n, m, k and d ($3 \le n \le 10^9$, $1 \le m, k \le 10^5$, $1 \le d \le 10^9$). The second line contains m space-separated integers c_i, which are the energy costs of a big jump for the corresponding frogs ($1 \le c_i \le 10^9$). The third line contains k space-seperated unique integers a_i, which are the indices of stones that must be visited exactly once ($2 \le a_i < n$).

Output

In the first and only line of the output print a single integer — minimum total energy cost.

Examples

standard input	standard output
10 2 3 3 4 7 4 8 7	4
10 2 2 3 4 7 9 5	15

Problem I. Slot Machine

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Zenyk wants to win a prize on a slot machine. Slot machine consists of N boxes. i-th box contains L_i balls, each ball has a color C_{ij}.

On each turn Zenyk pays one coin, chooses one box and gets one random ball from chosen box. He wins a prize if he gets two balls of the same color. Now Zenyk is interested what is the minimum number of coins he needs to pay to guarantee winning the prize. That means that for any sequence of balls he get on each turn he can obtain 2 balls of the same color. Note that Zenyk can decide which box to choose after previous turn.

Help Zenyk to find this number.

Input

First line of the input contains one integer N ($1 \leq N \leq 10^5$). Each of the next N lines contains integer L_i ($1 \leq L_i \leq 10^5$) followed by L_i integers C_{ij} – colors of the balls in the i-th box ($1 \leq C_{ij} \leq 10^5$).

It's guaranteed that there is at least one pair of balls with the same color, and that $\sum_{i=1}^{n} L_i \leq 10^5$.

Output

Print one number – minimum number of coins.

Example

standard input	standard output
7 4 1 2 3 4 1 1 1 2 1 3 1 4 7 4 7 4 4 7 7 4 1 5	2

Note

At first Zenyk chooses first box and then one of the boxes 2-5 depending on the color of the ball he get.

Problem J. Half is Good

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

— Hey Zenyk, would you please help me solve a problem?

— Everything for you, my dear!

— You are given an undirected weighted graph with unique weights and at most 10 million vertices and edges. You have to find a minimum spanning forest of the graph.

— Are you kidding me?! It's impossible under the given time and memory constraints!

— Alright, alright, calm down. Why don't you find at least half of edges that belog to a minimum spanning forest?

— Now we're talking!

Input

The first line of the input contains three integers n, m and s ($1 \le n, m \le 10^7$, $1 \le s \le 10^9$).

The edges of the graph must be generated in the following way:

```
unsigned s; // s in the value given in the input
unsigned getNext() {
    s = s xor (s << 13);
    s = s xor (s >> 17);
    s = s xor (s << 5);
    return s;
}
for (i = 0; i < m; ++i) {
    u = getNext() mod n;
    v = getNext() mod n;
    w = getNext() / 4;
    w = w * getNext(); // watch out for integer overflow
    // there is an undirected edge between u and v with weight w
}
```

Please note that vertices are numbered using 0-based indices. It's guaranteed that the weights of the edges are unique. The given graph may contain multi edges and loops.

Output

Print exactly $\lceil \frac{m}{32} \rceil$ 32-bit unsigned integers, where the j-th bit of i-th integer is set to 1 if and only if the edge with index $32 \times i + j$ is in your answer.

Example

standard input	standard output
4 7 47	72

Note

Minimum spanning forest of a graph is a subset of edges with minimum total weight, such that a pair of vertices is connected if and only if it's connected in the original graph. In other words, minimum spanning forest is the combination of minimum spanning trees of all connected components of the graph.

In the sample test case, the list of edges is the following:

```
1 1 179006535185096976
1 1 965163397507858962
2 2 41544785271292014
1 2 44839559531155752
2 1 2874637702340756660
1 3 2381022734547501765
3 2 1456859069025567641
```

The minimum spanning tree includes edges with indices 3 and 6 (0-based). Thus, answers $8(2^3)$, $64(2^6)$ and $72(2^3 + 2^6)$ are all considered correct.

Problem K. Dance

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

Marichka and her girlfriends are preparing a dance act for the upcoming New Year celebration. Their first task is to setup a starting lineup and divide into groups for the dance move.

There are n girls (including Marichka), and the i-th girl is initially located at point x_i on a line that goes from the left to the right. They can divide themselfs into as many groups as they like, in such a way that each girl belongs to exactly one group. They calculate ineffectiveness of j-th group as integer value $a + b \times (r_j - l_j)$, where l_j and r_j are the positions of the leftmost and the rightmost girls in the j-th group, respectively.

Since they don't like the current lineup, they decided that each girl will move excatly d units to the left or to the right. Note that multiple girls may be located at the same position.

They would like to know the minimum possible total ineffectiveness after all girls move and divide themselfs in groups.

Input

The first line contains four integers n, d, a and b ($1 \le n \le 100$, $1 \le d \le 50$, $1 \le a, b \le 10^6$). The second line contains n integers x_i, which are the initial positions of the girls ($1 \le x_i \le 100$).

Output

In the first and only line print the answer to the problem.

Examples

standard input	standard output
3 1 2 1 4 1 7	5
3 1 7 1 4 1 7	11

Problem L. Impress Her

Input file: *standard input*
Output file: *standard output*
Time limit: 1.5 seconds
Memory limit: 256 mebibytes

Zenyk decided to impress Marichka and solve the following interesting problem.

You are given a matrix of size $n \times m$ filled with integers. It's guaranteed that all cells which contain the same value are 4-side connected.

Let's define a convex hall of a connected component as minimum-area rectangle (with sides parallel to the matrix sides) that covers all cells of the component. The task is to count the number of pairs of components a and b for which the convex hall of a is inside the convex hall of b. Please note that the convex halls may touch by the side.

Input

The first line contains a pair of integers n and m ($1 \le n, m \le 500$) — the number of rows and columns of the matrix. The next n lines contain m integers each, which represent the matrix. It's guaranteed that matrix integers will be non-negative and won't exceed 10^6.

Output

In the only line print a single integer — the answer to the problem.

Example

standard input	standard output
3 4 1 2 2 4 1 1 1 4 5 1 7 4	3