Problem A. Modulo Ruins the Legend

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Grammy has a sequence of integers $a_1, a_2, \ldots, a_n$. She thinks that the elements in the sequence are too large, so she decided to add an arithmetic progression to the sequence. Formally, she can choose two non-negative integers s, d, and add $s + kd$ to a_k for each $k \in [1, n]$.

Since we want to ruin the legend, please tell her the minimum sum of elements modulo m after the operation. Note that you should minimize the sum **after** taking the modulo.

Input

The first line contains two integers n, m ($1 \le n \le 10^5$, $1 \le m \le 10^9$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < m$), denoting the initial sequence.

Output

Output exactly two lines.

The first line contains one integer, denoting the minimum sum of elements modulo m.

The second line contains two integers s, d ($0 \le s, d < m$), denoting the integers chosen by Grammy. **If there are multiple solutions, output any.**

Examples

standard input	standard output
6 24 1 1 4 5 1 4	1 0 5
7 29 1 9 1 9 8 1 0	0 0 0

Problem B. Useful Algorithm

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	1024 megabytes

Putata is learning some useful algorithm called Binary Adding these days. This algorithm allows you to calculate the sum of two m-bit binary integers.

An integer x is called to be a m-bit binary integer, if and only if $0 \le x < 2^m$. The binary representation of m is a 0-indexed sequence v of length m, where $\forall 0 \le i < m, v_i \in \{0, 1\}$, and $x = \sum\limits_{i=0}^{m-1} v_i 2^i$. It is guaranteed that for all m-bit binary integers, each one have its unique binary representation.

The Binary Adding algorithm for calculating the sum of two m-bit binary integers with binary representation $\{a_i\}_{i=0}^{m-1}, \{b_i\}_{i=0}^{m-1}$ is shown below:

Algorithm 1 BinaryAdding(a, b)

Input: input parameters $a[0, \dots, m-1]$, $b[0, \dots, m-1]$
Output: output result $sum[0, \dots, m], carry[0, \dots, m]$

$carry[0] := 0$
for $i := 0$ to $m - 1$ **do**
    $sum[i] := a[i] \oplus b[i] \oplus carry[i]$ ▷ $\oplus$ denotes xor operation
    $carry[i+1] := (a[i] \wedge b[i]) \vee (a[i] \wedge carry[i]) \vee (b[i] \wedge carry[i])$ ▷ $\wedge$ denotes and operation, $\vee$ denotes or operation.
$sum[m] := carry[m]$
return $sum[0, \dots, m], carry[0, \dots, m]$

In order to test if Putata really mastered this algorithm, Budada is going to prepare tests for Putata. Before preparing his tests, Budada devised a way to calculate the difficulty of a problem. Assume the problem is to calculate the sum of $\{a_i\}_{i=0}^{m-1}, \{b_i\}_{i=0}^{m-1}$, then we define the carry set of a, b, $S(a, b) = \{x | \text{when calling BinaryAdding}(a, b), carry_x = 1\}$. Budada has an integer sequence $\{w_i\}_{i=1}^{m}$, denoting the difficulty of calculation when a carry occurred at the corresponding bit. The **Carry Difficulty** is the maximum difficulty among all bits where a carry occurred. If there's no carry occurred, the **Carry Difficulty** is 0.

The problem database of Budada is an integer sequence $\{c_i\}_{i=1}^{n}$. Each integer has a corresponding **Numerical difficulty**, which is also an integer sequence $\{d_i\}_{i=1}^{n}$. Please notice that c_i are **not necessarily** pairwise distinct, and for some $c_i = c_j$, d_i **might not be equal to** d_j. A Binary Adding problem consists of two integers, so when Budada chooses c_i and c_j to set a test, the **Numerical difficulty** of this problem is $d_i + d_j$.

Budada wants to prepare the most difficult test for Putata, so he will choose two integers i, j such that $1 \le i, j \le n$ (not necessarily distinct), and use c_i, c_j to set a test. He wants to maximize the **Test Difficulty**, which is the product of the **Carry Difficulty** and **Numerical difficulty** of this test, and he asked you to tell him this production. Formally, the maximum **Test Difficulty** is $\max\limits_{1 \le i,j \le n} \{\max\{\max\{w_x | x \in S(c_i, c_j)\}, 0\} \cdot (d_i + d_j)\}$.

Budada also has q updates for his problem database, each time he will select an integer i, and modify c_i, d_i. You are asked to answer the maximum **Test Difficulty** of $q + 1$ versions of the problem database.

Input

The first line contains three integers n, m, q ($1 \le n, q \le 10^5$, $1 \le m \le 16$), corresponding to the meaning described above.

The second line contains m integers, the i-th integer is w_i $(1 \le w_i \le 10^9)$.

The third line contains n integers, the i-th integer is c_i $(0 \le c_i < 2^m)$.

The fourth line contains n integers, the i-th integer is d_i $(1 \le d_i \le 10^9)$.

For the following q lines, each line contains three integers x, u, v, let $lastans$ denotes the maximum **Test Difficulty** of the last version of the problem database, and $x' = x \oplus lastans$, $u' = u \oplus lastans$, $v' = v \oplus lastans$ $(1 \le x' \le n,\ 0 \le u' < 2^m,\ 1 \le v' \le 10^9)$, this represents Budada changes $c_{x'}$ to u', and $d_{x'}$ to v'. Here, "$\oplus$" denotes the bitwise XOR operator. **Please notice that you should use 64-bit integer to store x, u, v.**

Output

Output $q+1$ lines, denoting the maximum **Test Difficulty** of the $q+1$ versions of the problem database.

Example

standard input	standard output
5 3 3 1 2 4 0 0 1 2 7 10 10 5 3 1 27 24 29 20 16 19 13 8 9	24 16 8 0

Note

The decrypted operations are:

$x' = 3, u' = 0, v' = 5.$

$x' = 4, u' = 0, v' = 3.$

$x' = 5, u' = 0, v' = 1.$

Problem C. No Bug No Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Putata is preparing the RPG Pro League (RPL) held by the International Computer Playing Company (ICPC). In this RPG game, the player will wear n items at the same time. Each item can offer the player several points of power. There is a magic buff in the game, which can upgrade each item such that they can offer several points of bonus power.

However, the buff is limited, it can only buff at most k points of power. Formally, assume the player is wearing nothing initially, and then will wear all the n items one by one. The game server will scan through all these n items one by one, according to the permutation that the player wears them. When the server is scanning the i-th item, which can offer p_i points of power, let $sum = \sum_{1 \leq j < i} p_j$ denoting the total points of power scanned before:

- If $sum + p_i \leq k$, the whole item will be upgraded. The buff will offer w_{i,p_i} points of bonus power.
- If $sum \geq k$, the item won't be upgraded. The buff will offer nothing.
- Otherwise, only a part of the item will be upgraded. The buff will offer $w_{i,k-sum}$ points of bonus power.

Putata is clever, he soon realized that he can adjust the permutation to wear these n items to gain more points of bonus power! Unfortunately, Putata doesn't know the optimal permutation, please write a program to help him.

The behavior of the magic buff performed by the game server is a bug. The game code can work all thanks to bugs, so it is possible that $w_{i,a} > w_{i,b}$ where $a < b$.

Input

The first line contains two integers n and k ($1 \leq n \leq 3\,000$, $0 \leq k \leq 3\,000$), denoting the number of items and the limit k.

Each of the following n lines starts with an integer p_i ($1 \leq p_i \leq 10$), denoting the base power of the i-th item, followed by p_i integers $w_{i,1}, w_{i,2}, \ldots, w_{i,p_i}$ ($1 \leq w_{i,j} \leq 10^5$).

Output

Output a single line containing an integer, denoting the maximum points of total bonus power that can be reached. The base power is not included in the answer.

Example

standard input	standard output
4 5 2 1 3 2 1 1 2 3 1 2 1 3	9

Problem D. Money Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Putata and Budada are organizing a game with n players. Each player has some deposit, which is a real number. Player i has a_i deposits in the beginning. During each round of the game, the followings happen **in order**:

- Player 1 gives player 2 half of player 1's deposit.
- Player 2 gives player 3 half of player 2's deposit.
- ...
- Player $n-1$ gives player n half of player $n-1$'s deposit.
- Player n gives player 1 half of player n's deposit.

The n players played this game for exactly 2022^{1204} rounds. Putata wonders how much deposit each player has after the game. Please write a program to answer his question.

Input

The first line contains an integer n ($2 \le n \le 10^5$), denoting the number of players.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$), denoting the deposit player i has in the beginning.

Output

Output one line with n real numbers, denoting the deposit each player has after they played this game.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}. Formally, let your answer be a, and the jury's answer be b. Your answer will be considered correct if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

Example

standard input	standard output
2 4 2	4.00 2.00

Note

During one round, the deposit they have changed as follows: $[4, 2] \to [2, 4] \to [4, 2]$. Their deposit does not change after this round, so the answer is the same as the input.

Problem E. Oscar is All You Need

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Putata has a sequence p of length n, where p is a permutation of $1, 2, \ldots, n$. Budada can perform the following operation at most $2n + 1$ times:

- Cut the sequence into three consecutive non-empty parts, and swap the first part and the last part. Formally, you should select two integers x, y satisfying that $x > 0$, $y > 0$, $x + y < n$, and the sequence will change from $p_1, \ldots, p_x, p_{x+1}, \ldots, p_{n-y}, p_{n-y+1}, \ldots, p_n$ to $p_{n-y+1}, \ldots, p_n, p_{x+1}, \ldots, p_{n-y}, p_1, \ldots, p_x$.

Budada wants to make the lexicographical order of the permutation as small as possible after no more than $2n + 1$ operations. Please help him find the way to perform operations so that the lexicographical order of the permutation is as small as possible.

A **permutation** is an array where each integer from 1 to s (where s is the size of permutation) occurs exactly once.

A permutation a is lexicographically smaller than a permutation b if and only if the following condition holds:

- Let x be the smallest integer where $a_y = b_y$ holds for all $y \le x$, then we have $x < n$ and $a_{x+1} < b_{x+1}$.

Input

The input contains several test cases.

The first line contains an integer T ($1 \le T \le 120$), denoting the number of test cases.

For each test case, the first line contains an integer n ($3 \le n \le 1000$), denoting the length of the permutation.

The second line contains n integers, the i-th integer is p_i ($1 \le p_i \le n$), denoting the permutation. It is guaranteed that p is a permutation of $1, 2, \ldots, n$.

It is guaranteed that the sum of n in all test cases will not exceed 1000.

Output

For each test case, output one integer m in the first line, denoting the number of operations. You should guarantee that $0 \le m \le 2n + 1$.

Then output m lines, each line contains two integers x, y, denoting one operation. You should guarantee that $0 < x$, $0 < y$, $x + y < n$.

Please notice that you **do not have to** minimize the number of operations.

Example

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

Problem F. Da Mi Lao Shi Ai Kan De

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	1024 megabytes

Grammy has joined $n + 1$ QQ groups numbered $0 \sim n$. Teacher Rice is in the group 0.

Every day, Grammy's friends send messages to some groups in $1 \sim n$, and Grammy will select the messages which Teacher Rice likes and forward them to the group 0.

Here we define a message as a string consisting of lowercase letters, and Teacher Rice likes a message if and only if the string "`bie`" is a substring of the message.

Now, given the messages in groups $1 \sim n$, Grammy will search messages from group 1 to group n in order, checking the messages in the group one by one. For each message, pick out it if Teacher Rice likes it and it hasn't appeared in group 0, and forward it to group 0. Here appear means the same message has been forwarded to group 0 before.

Please output all the messages that Grammy will forward to group 0 in order. For each group, if Grammy can't pick out any message, output "`Time to play Genshin Impact, Teacher Rice!`" in one line.

Input

The first line contains a single integer n ($1 \leq n \leq 10^4$), denoting the number of QQ groups. The following lines describe the messages in groups 1 to n.

For the i-th group, the first line contains an integer m_i ($0 \leq m_i \leq 10^4$), denoting the number of messages in this group. The following m_i lines each line contains a non-empty string $s_{i,j}$ representing a message.

It is guaranteed that $\sum m_i \leq 10^4$, $\sum |s_{i,j}| \leq 10^4$ and all the messages only consist of lowercase letters.

Output

Output each message in a line, denoting all the messages that Grammy will forward to group 0 in order. For each group, if Grammy can't pick out any message, output "`Time to play Genshin Impact, Teacher Rice!`" in one line.

Example

standard input

```
6
1
biebie
1
adwlknafdoaihfawofd
3
ap
ql
biebie
2
pbpbpbpbpbpbpbpb
bbbbbbbbbbie
0
3
abie
bbie
cbie
```

standard output

```
biebie
Time to play Genshin Impact, Teacher Rice!
Time to play Genshin Impact, Teacher Rice!
bbbbbbbbbbie
Time to play Genshin Impact, Teacher Rice!
abie
bbie
cbie
```

Problem G. Subgraph Isomorphism

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

Grammy wants to get the Turing Award! She decided to solve the Subgraph Isomorphism problem in polynomial time.

Since the problem is indeed too hard, she begins with doing some simplifications and trying to solve the simplified problem first.

Now Grammy has a connected undirected graph G with n vertices and m edges. She wants to know whether it is possible to find a tree T with n vertices such that all connected subgraphs of G with n vertices and $n-1$ edges are isomorphic to T. Grammy knows the answer for sure, but she wants to give you a quiz.

Two graphs G and H are isomorphic if and only if there exists a bijection between the vertex sets of G and H ($f : V(G) \to V(H)$) such that any two vertices u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H.

Two vertices are adjacent if and only if they are directly connected by an edge.

Input

The input consists of multiple test cases.

The first line contains an integer T ($1 \le T \le 10^5$), denoting the number of test cases.

For each test case, the first line contains two integers n, m ($1 \le n \le 10^5$, $n-1 \le m \le 10^5$), denoting the number of vertices and the number of edges respectively.

Each of the next m lines contains two integers u_i, v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), denoting an edge (u_i, v_i).

It is guaranteed that there are no multiple edges and the graph is connected.

It is guaranteed that the sum of n and the sum of m in all test cases will not exceed 10^6.

Output

For each test case, output one line containing either "`YES`" or "`NO`".

Example

standard input	standard output
4 7 6 1 2 2 3 3 4 4 5 5 6 3 7 3 3 1 2 2 3 3 1 5 5 1 2 2 3 3 4 4 1 1 5 1 0	YES YES NO YES

Problem H. RPG Pro League

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The International Computer Playing Company (ICPC) is recently scheduling the annual RPG Pro League (RPL). In the RPG game, there are three kinds of different roles: Damager, Synergier, and Buffer. A team consists of exactly four players. Only the following two types of teams are allowed in RPL:

- One Damager, two Synergiers, and one Buffer.
- Two Damagers, one Synergier, and one Buffer.

Before the real competition, the ICPC decides to hold an exhibition game. There are n players, labeled by $1, 2, \ldots, n$. The i-th player can only play roles in set S_i, and the price to invite him to participate in the exhibition game is p_i dollars.

You are working for the ICPC. Your job is to select which players to invite such that they can make the maximum number of valid teams, and the total cost is minimized. Note that a player can not join multiple teams.

Unfortunately, the players may always adjust their prices. You will be given q events, in each event, the price of a player will be changed. Your task is to report the current minimum total cost for maximizing the valid teams after each event.

Input

The first line contains a single integer n ($1 \le n \le 10^5$), denoting the number of players.

Each of the following n lines contains a string S_i and an integer p_i ($1 \le |S_i| \le 3$, $1 \le p_i \le 10^9$), denoting the role set and the price of the i-th player. S_i consists of at most three different characters in {'D', 'S', 'B'}, denoting Damager, Synergier, and Buffer, respectively.

The next line contains a single integer q ($1 \le q \le 10^5$), denoting the number of events.

Each of the following q lines contains two integers x_i and y_i ($1 \le x_i \le n$, $1 \le y_i \le 10^9$), denoting the price of the x_i-th player is changed into y_i dollars.

Output

Output q lines, the i-th ($1 \le i \le q$) of which contains an integer denoting the minimum total cost for maximizing the valid teams after the i-th event.

Example

standard input	standard output
5 BS 3 D 3 B 2 D 4 D 5 2 5 2 1 5	10 12

Problem I. Guess Cycle Length

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

This is an interactive problem.

Grammy has a directed cyclic graph of n vertices ($1 \le n \le 10^9$) numbered from 1 to n. A directed cyclic graph is a directed graph of n vertices that form one cycle. Specifically, there are n vertices and n edges in the graph, and there exists a permutation $p_1, p_2, \ldots, p_n$ such that there is a one-way edge from p_i to $p_{(i \bmod n)+1}$.

Initially, there is a token on a predetermined vertex.

You can ask Grammy to move the token in the following way:

"`walk x`" where $0 \le x \le 10^9$. In response to the query, Grammy will move the token through exactly x edges and tell you the position of the token after moving.

You win if you guess the number of vertices in the hidden graph (number n) by making no more than 10^4 queries.

The vertices in the graph and the initial position of the token are fixed in advance.

Interaction Protocol

You can make no more than 10^4 queries. To make a query, output "`walk x`" ($0 \le x \le 10^9$) on a separate line, then you should read the response from standard input.

In response to the query, the interactor will move the token through exactly x edges and output the position of the token after moving.

To give your answer, print "`guess n`" on a separate line, where n is the size of the hidden graph ($1 \le n \le 10^9$). The output of the answer is **not** counted towards the limit of 10^4 queries.

After that, your program should terminate.

After printing a query, do not forget to output end of line and flush the output. To do this, use `fflush(stdout)` or `cout.flush()` in C++, `System.out.flush()` in Java, `flush(output)` in Pascal, or `stdout.flush()` in Python.

It is guaranteed that the vertices in the graph and the initial position of the token are fixed in advance.

Example

standard input	standard output
	walk 0
3	
	walk 1
10	
	walk 2
4	
	walk 3
5	
	walk 4
3	
	walk 6
5	
	guess 10

Problem J. Painting

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

Putata likes painting very much. He is now painting on a piece of white paper, which can be regarded as a 2D plane. Initially, Putata drew a straight line $x = W$ on the paper. In each of the next n steps, Putata will draw a segment. He will start drawing the i-th segment from $(0, a_i)$ to (W, b_i). If his pencil touches any other segment drawn before, he will stop drawing at the point he touches other segments. After drawing a segment, Putata may think the current figure is not so beautiful, and erase the segment he just drew.

In this problem, your task is to report where each segment will end at.

Input

The first line contains two integers n and W ($1 \le n \le 3 \times 10^5$, $1 \le W \le 10^6$), denoting the number of segments and the parameter W.

Each of the following n lines contains three integers a_i, b_i and c_i ($1 \le a_i, b_i \le 10^6$, $0 \le c_i \le 1$). Putata will erase the i-th segment if and only if $c_i = 0$. It is guaranteed that $(0, a_i)$ will not coincide with other segments.

Output

Output n lines, the k-th ($1 \le k \le n$) of which contains the coordinate $(u_1/v_1, u_2/v_2)$ where the k-th segment will end at. You should guarantee that $\gcd(u_1, v_1) = \gcd(u_2, v_2) = 1$.

Example

standard input	standard output
4 3 1 2 1 2 1 1 3 1 0 3 2 1	(3/1,2/1) (3/2,3/2) (2/1,5/3) (3/1,2/1)

Problem K. Master of Both

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Professor Hui-Bot is the master of string theory and advanced data structures, so he came up with an interesting problem. Given a sequence of n strings consisting of only lowercase English letters, how many inversions are there in this sequence when the strings are compared by lexicographical order?

As the most extraordinary student of Hui-Bot, Putata and Budada mastered superb string theory and advanced data structure skills respectively, and they solved this problem together with ease. However, there are q different parallel universes, where the characters in the alphabet are not appearing in the original order.

Formally, the alphabet in each universe is a string, which is a permutation of the 26 lowercase English letter, denoting the order each character appears.

A string a is lexicographically smaller than a string b if and only if one of the following holds:

- a is a prefix of b, but $a \neq b$;
- in the first position where a and b differ, the string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

The number of inversions in a sequence a of length n is the number of ordered pairs (i, j) such that $1 \leq i < j \leq n$, $a_j < a_i$.

Please help Putata and Budada in each universe to solve the problem.

Input

The first line of the input contains two integers n, q ($1 \leq n \leq 5 \times 10^5$, $1 \leq q \leq 5 \times 10^4$), denoting the length of the sequence.

For the following n lines, the i-th line contains a string s_i ($1 \leq |s_i| \leq 10^6$). It is guaranteed that the string consists of only lowercase English letters, and $\sum_{i=1}^{n} |s_i| \leq 10^6$.

For the following q lines, each line contains a string t, denoting the alphabet in one universe. It is guaranteed that t is a permutation of 26 lowercase English letters.

Output

Output q lines, denoting the answer in q universes.

Example

standard input	standard output
5 3 aac oiputata aaa suikabudada aba abcdefghijklmnopqrstuvwxyz qwertyuiopasdfghjklzxcvbnm aquickbrownfxjmpsvethlzydg	4 3 4

Problem L. Levenshtein Distance

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

The **Levenshtein Distance** between two strings is the smallest number of simple one-letter operations needed to change one string to the other. The operations are:

- Adding a letter anywhere in the string.
- Removing a letter from anywhere in the string.
- Changing any letter in the string to any other letter.

You will be given a number k and two strings S and T. Your task is to find the number of non-empty substrings of T whose Levenshtein Distance between S is exactly i for every possible non-negative integer i ($0 \le i \le k$). Two substrings are considered different if and only if they occur in different places.

Input

The first line contains a single integer k ($0 \le k \le 30$), denoting the parameter k.

The second line contains a string S ($1 \le |S| \le 10^5$), denoting the pattern string.

The third line contains a string T ($1 \le |T| \le 10^5$), denoting the text string.

It is guaranteed that the input strings only consist of lowercase English letters ('`a`' to '`z`'), uppercase English letters ('`A`' to '`Z`'), and digits ('`0`' to '`9`').

Output

Output $k+1$ lines, the i-th ($1 \le i \le k+1$) of which contains an integer denoting the number of substrings of T whose Levenshtein Distance between S is exactly $i-1$.

Example

standard input	standard output
4 aaa aabbaab	0 5 15 7 1

Problem M. Please Save Pigeland

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

Some horrible disease called Mysterious Oscar is spreading in Pigeland. The Pigeland has n cities and is connected by $n-1$ bi-directional roads. The i-th road is connecting city u_i and city v_i, with length w_i. It is guaranteed that for every city it is possible to travel to any other city using these roads.

Now, there are k distinct cities $c_1, c_2, \ldots, c_k$ having the horrible disease. As the leader of the Pigeland, Putata and Budada are going to save the Pigeland. First they will find a city r to build up a hospital, without considering whether city r is infected by the disease. Then, they will use the rest of their money to build the only vehicle which could travel in Pigeland, the Pigpigcar. Because they are strapped for cash, they can only build one Pigpigcar, and once a Pigpigcar is built, some parameter d of the Pigpigcar is set, and could not be changed. The Pigpigcar having parameter d can only travel from one city to the other, where the distance between the two cities should be a multiple of d. Formally, if you want to travel from city u to city v, the distance of the shortest path between u, v should be $p \times d$, where p is a non-negative integer, and this would cost p Pigecoins. Please notice that if you want to travel from city u to city v, it is **not necessary** to stop at every city on the path from u to v, the Pigpigcar can directly go from u to v, if the minimum distance between u and v is a multiple of d.

For the following k days, Putata and Budada will take the following actions to save the Pigeland. On the i-th day, they will travel to city c_i from city r **along the shortest path** between city r and city c_i using the Pigpigcar, cure all the piggies in the city and then travel back to city r from city c_i.

Putata and Budada want you to choose the proper city r to build the hospital, and the parameter d of the Pigpigcar they should build, so that the Pigecoins spent during the travel is minimized. Please help them to find the minimum number of Pigecoins spent.

Input

The first line contains two integers n, k ($1 \le n \le 5 \times 10^5$, $1 \le k \le n$), denoting the number of cities and the number of cities having disease.

The second line contains k distinct integers $c_1, c_2, \ldots, c_k$ ($1 \le c_i \le n$), denoting the cities having disease.

For the following $n-1$ lines, each line contains three integers u, v, w ($1 \le u, v \le n$, $u \ne v$, $1 \le w \le 10^7$), denoting a road connecting city u and v with length w.

It is guaranteed that you can go from any city to any other city using these roads.

Output

Output one integer in one line, denoting the minimum number of Pigecoins spent.

Example

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

Note

In the sample, you should choose city 1 to build hospital and build a Pigpigcar with parameter 6, the distance between city 1 and city $3, 4, 5$ is $6, 12, 6$, so the total Pigcoins costed is $1 \times 2 + 2 \times 2 + 1 \times 2 = 8$.