The 44th International Collegiate Programming Contest Asia Nanchang Regional Contest

November 10

江西师范大学
JIANGXI NORMAL UNIVERSITY

Asia Regional 2019

icpc International Collegiate Programming Contest

Problems

Do not open before the contest has started.

Problem A. 9102

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

As everyone knows, there are innumerable "world lines" representing different worlds. Every tiny change maybe creates a new world line.

It is 9102 A.D. now and human beings have lived on Mars for thousands of years, forming a lot of tribes. Every tribe contains one or more families. Two tribes can go to integration, which all families in one tribe move to the other; One family may become extinct or migrate from their original tribe to another family's tribe.

You are ruling the universe as Creator so that you know everything on Mars. Because of the reason that you usually forget things, you decide to write a program to manage these happening things.

Suppose that there are n tribes on Mars and at the beginning, each tribe consists of the unique family. We number these tribes as well as these families from 1 to n. Your program needs to support the following five types of operations:

- "`1 k a b`": merge the tribe containing the a-th family and the tribe containing the b-th family;
- "`2 k a`": make the a-th family into extermination;
- "`3 k a b`": the a-th family moves away from their tribe and join in the tribe containing the b-th family;
- "`4 k a b`": report that if the a-th family and the b-th one belong to the same tribe;
- "`5 k a`": report the total number of families in the tribe which contains the a-th family.

Note that for each given operation with a parameter k, you are asked to consider the state after the k-th operation.

It is possible that the a-th family and the b-th one are in the same tribe when you deal with an operation of type 1 or 3, then nothing will happen. It is also possible that given numbers a or b indicates to an exterminated family, then nothing will happen for the first three types of operations, and the report should be "`No`" for the fourth type of operation, and "`0`" for the fifth one;

Input

The first line contains two numbers n and m ($1 \leq n, m \leq 1000000$), representing the initial number of families and operations.

The next m lines contain five kinds of operations. The i-th line contains three or four numbers op_i, k_i, a_i and b_i ($1 \leq op_i \leq 5, 0 \leq k_i \leq i - 1, 1 \leq a_i, b_i \leq n$).

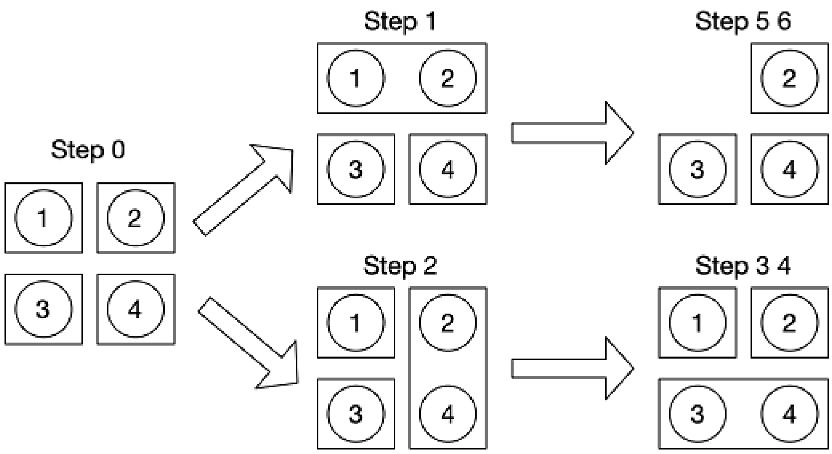
Output

For operation 4, you need to output one line "`Yes`" or "`No`" (without quotes); for operation 5, you need to output one line contains a number.

Example

standard input	standard output
4 6 1 0 1 2 1 0 4 2 3 2 4 3 5 3 3 2 1 1 4 5 3 4	2 No

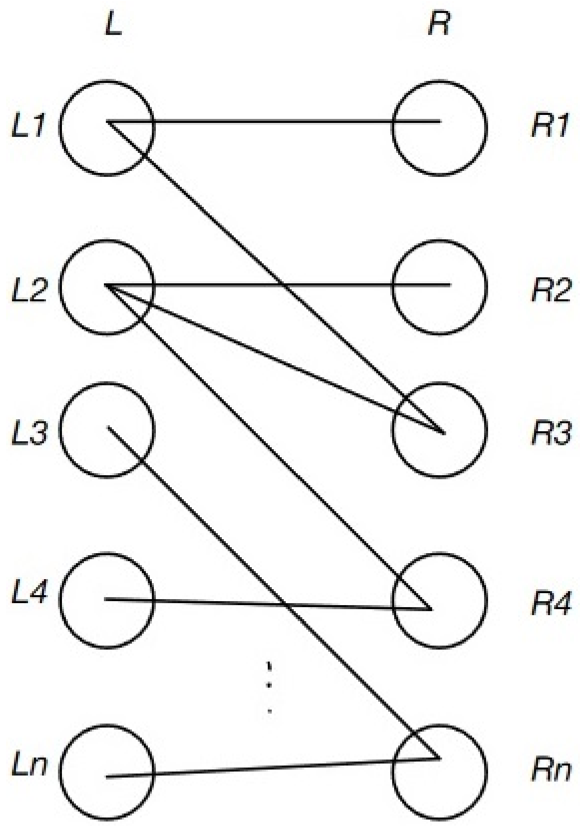
Note



Problem B. A Funny Bipartite Graph

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Bob is really interested in bipartite graph, and he needs help for this problem:



Bob has a bipartite graph which meets all the conditions below:

- The left part has n $(1 \le n \le 18)$ vertices, named $L_1, L_2, \ldots, L_n$. And the right part also has n vertices, named $R_1, R_2, \ldots, R_n$.
- L_i **never** connects with R_j when $i > j$ for all pairs of (i, j) that $1 \le i, j \le n$.
- For each vertex on the left part L_i $(1 \le i \le n)$, its degree D_i is at most 3 and at least 1 $(1 \le D_i \le 3)$.

Bob wants to select some edges from the graph such that the selected edges and their endpoints form a new graph. However, he should obey these rules:

- Some pairs of vertices can't appear in the new graph together. You are given an $n \times n$ matrix A, $A_{i,j}$ is 1 if and only if L_i and L_j can't appear in the new graph together.
- Bob must ensure that the selected edges cover all the right vertices. In other words, every right vertex should appear in the new graph.
- Every left vertex L_i $(1 \le i \le n)$ has a magic number M_i $(1 \le M_i \le 100)$. If L_i appears in the new graph, its cost is ${M_i}^{d_i}$, where d_i is the degree of L_i in the new graph, otherwise its cost is zero.

Now Bob wants to select edges satisfying the rules above with minimum total cost. Please write a program to help him find the minimum total cost, or determine it is impossible to get such a valid new graph.

Input

There are several test cases in the input file.The first line contains one integer T $(1 \leq T \leq 10)$ — number of the test cases. For each case:

The first line contains one integer n $(1 \leq n \leq 18)$ — number of the left vertices.

The following n lines are about the graph edges information with an $n \times n$ matrix G. The i-th line contains a 01-string G_i and the j-th character $G_{i,j}$ is 1 if and only if L_i and R_j are connected, otherwise $G_{i,j}$ is 0. We guarantee that the degree of each left vertex L_i is between 1 and 3. We also guarantee that $G_{i,j}$ is always 0 when $i > j$.

The following line is empty.

The following n lines are about the constraints on left vertices with an $n \times n$ matrix A. The i-th line contains a 01-string A_i and the j-th character $A_{i,j}$ is 1 if and only if L_i and L_j can't be in S together. We guarantee that $A_{i,i}$ $(1 \leq i \leq n)$ is 0 and $A_{i,j} = A_{j,i}$ $(1 \leq i, j \leq n)$.

The following line is about magic numbers containing n integers. The i-th integer M_i $(1 \leq M_i \leq 100)$ is the magic number of L_i.

Output

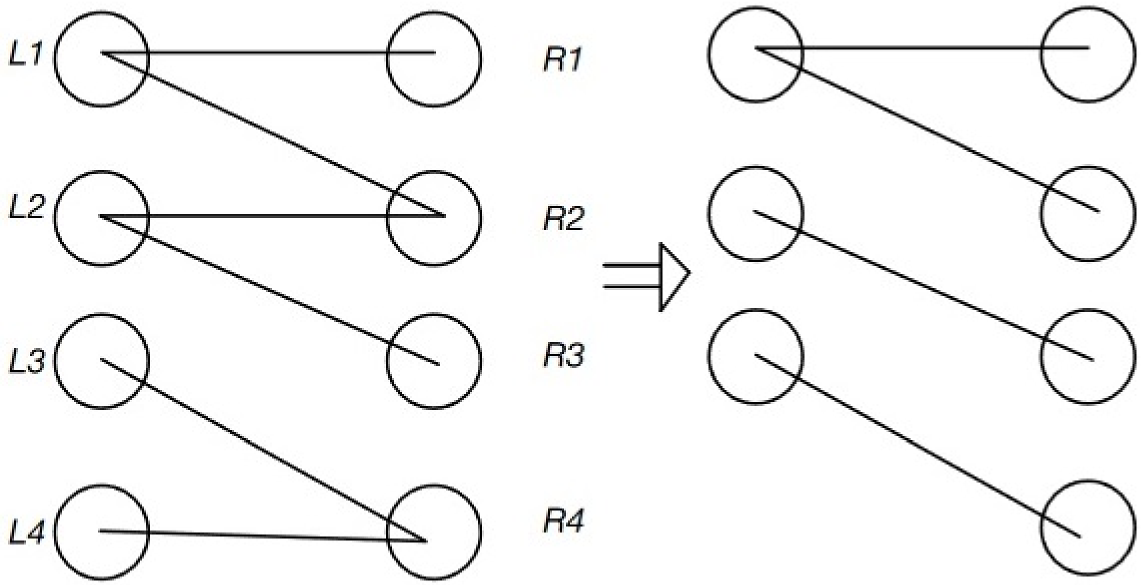
For evey case print a single line containing an integer, the minimum total cost. If it is impossible to get such a valid new graph, just output "`-1`" instead.

Example

standard input	standard output
2 4 1100 0110 0001 0001 0001 0001 0000 1100 3 4 4 5 4 1100 0110 0001 0001 0011 0001 1000 1100 3 4 4 5	17 -1

Note

In the first case:



We can choose edges $(L_1, R_1), (L_1, R_2), (L_2, R_3), (L_3, R_4)$ such that $S = \{L_1, L_2, L_3\}$, satisfies the A matrix's constraints. And the degrees are $d_1 = 2, d_2 = 1, d_3 = 1$, so the total cost $= 3^2 + 4^1 + 4^1 + 0 = 17$.

In the second case, R_3 and R_4 can't be both covered at the same time, so there is no solution.

Problem C. And and Pair

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Given an extremely large non-negative integer n, you are asked to count the number of pairs (i, j) of integers satisfying the following conditions:

- $0 \le j \le i \le n$;
- $i \,\&\, n = i$; and
- $i \,\&\, j = 0$.

Here '`&`' represents the bitwise AND operator.

For simplicity, the binary representation of n will be given. Meanwhile, you only need to output the answer modulo $(10^9 + 7)$.

Input

The first line contains an integer T $(1 \le T \le 20)$ indicating the number of test cases.

Each of the following T lines contains a string S $(1 \le |S| \le 10^5)$ which is the binary representation of the non-negative integer n. Note that leading zeros of S could appear in the input.

Output

For each test case, output a line containing the answer described above modulo $(10^9 + 7)$.

Example

standard input	standard output
2 111 1010	14 15

Problem D. Bitwise Tree

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

You are given a tree with n nodes labelled from 1 to n, the root of which is the node labelled 1. Every node in the tree will be endowed with values related to given parameters $a_1, a_2, \cdots, a_n$, and each edge in the tree is endowed with a given bitwise operator which is one of the bitwise OR '`|`', the bitwise AND '`&`' and the bitwise XOR '`^`'.

For any pair of nodes (u, v) indicating the u-th and the v-th node in the tree, the shortest path connecting them in the tree, denoted by

$$u = p_0, p_1, p_2, \cdots, p_{s-1}, p_s = v$$

together with operators along the path provide a value associated with (u, v) given by

$$((a_{p_0}\ opt_1\ a_{p_1})\ opt_2\ a_{p_2}) \cdots opt_s\ a_{p_s})$$

where opt_i is the operator on the edge between the node p_{i-1} and the node p_i in this path.

Now you are given q queries, each of which contains two integers d and u $(1 \le u \le n)$. For this query, the value of the i-th node will be set to be $(a_i + i \times d)$, and you are asked to calculate the bitwise OR, AND and XOR sum respectively of values for all the pairs (u, v) where the u-th node is an ancestor of the v-th node. Note that a node is not considered as an ancestor of itself.

Input

The first line contains two integers n $(2 \le n \le 10^5)$ and q $(1 \le q \le 10^5)$ indicating the size of the tree and the number of queries.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^{18})$ descirbed as above.

The following $n - 1$ lines describe the tree. The i-th line of them contains two integers f $(1 \le f \le n)$ and s $(0 \le s \le 2)$ which describe that the f-th node is the parent node of the $(i + 1)$-th node, and the edge connecting the $(i + 1)$-th node and the f-th node is endowed with a bitwise operator: if $s = 0$ the operator is '`|`'; if $s = 1$ the operator is '`&`'; and if $s = 2$ the operator is '`^`'.

Each of the following q lines contains two integers d $(0 \le d \le 100)$ and u $(1 \le u \le n)$ describing a query. It is guaranteed that the node u is not a leaf in the tree.

Output

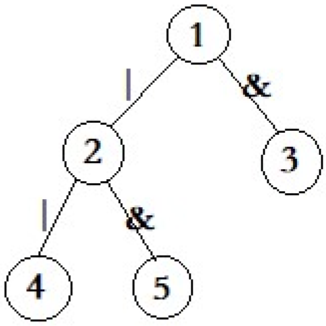
Output q lines corresponding to q given queries, the i-th of which contains three integers representing the bitwise OR sum, the bitwise AND sum and the bitwise XOR sum for the i-th query respectively.

Example

standard input	standard output
5 4 1 2 3 4 5 1 0 1 1 2 0 2 1 0 1 0 2 1 2 2 2	7 1 4 6 0 6 12 0 12 14 6 8

Note

In the sample case, the tree looks like



For the first query, the bitwise OR sum is $(1|2) \mid ((1|2)|4) \mid ((1|2)\&5) \mid (1\&3) = 7$.

For the second query, the bitwise OR sum is $(2|4) \mid (2\&5) = 6$.

Problem E. Bob's Problem

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Bob was in trouble. He rubbed the magic ring on his finger, and you came out of the ground.

You are given an undirected graph G which contains n vertices labelled from 1 to n, with m weighted edges between them coloured in black or white. You have to choose some edges in G such that there is at least one path between any two vertices only passing by selected edges, and you can select no more than k white edges. There may be multiple available strategies to determine these edges, and you are asked to find out the way with a maximum total weight of edges.

Input

The first line contains an integer T $(1 \leq T \leq 5)$ indicating the number of test cases.

For each test case, the first line contains three integers n $(1 \leq n \leq 50000), m$ and k $(1 \leq k \leq m \leq 500000)$.

Each of the following m lines contains four integers u, v $(1 \leq u, v \leq n), w$ $(0 \leq w \leq 100000)$ and c $(0 \leq c \leq 1)$ describing an edge of weight w and colour c between the u-th vertex and the v-th vertex. Here an edge is white if $c = 1$, or black if $c = 0$.

Note that there may be multiple edges between some vertices, and self-loops are also allowed.

Output

For each test case, output a single line with an integer indicating the maximum total weight of selected edges, or output "`-1`" if there is no solution for the given graph.

Example

standard input	standard output
1 5 6 2 1 2 0 0 1 3 5 1 1 5 1 0 2 3 6 1 2 4 2 0 3 4 7 1	16

Problem F. Dynamic Suffix Array

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

In computer science, a suffix array is a sorted array of all suffixes of a string. It is a data structure used, among others, in full-text indices, data compression algorithms and within the field of bibliometrics.

Assume $S[1, n]$ is a string, let's denote $S[i, j]$ as the substring of S ranging from i to j and denote suffix i as $S[i, n]$.

The suffix array A of S is now defined to be an array of integers providing the starting positions of suffixes of S in lexicographical order. This means, an entry $A[i]$ contains the starting position of the i-th smallest suffix in S and thus $S[A[i-1], n] < S[A[i], n]$ always holds for each $1 < i \leq n$.

Now there is an empty string S, you need to maintain its suffix array dynamically. There will be q events of the two kinds below:

- "`1 c`" : Append a character c at the end of S, it is guaranteed that c is chosen uniformly at random from lower-case english letters '`a`' to '`z`'.
- "`2 k`" ($1 \leq k \leq |S|$): You need to report the position of k in the current suffix array of S, which means you need to report the rank of suffix k.

Please write a program to process these events efficiently.

Input

The first line of the input contains an integer q ($2 \leq q \leq 1000000$), denoting the number of events.

For the next q lines, each line describes an event. It is guaranteed that the first event is of the first type, and the last event is of the second type.

Output

For each event of the second type, output an integer denotes the answer in a single line.

Example

standard input	standard output
10 1 t 1 s 2 2 1 r 1 w 1 f 1 t 2 2 1 q 2 2	1 3 4

Problem G. Eating Plan

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Bob is hungry now and he needs to eat some food. Alice puts n dishes of food in front of him, numbered from 1 to n. Alice tells him that he can only eat continuous dishes of food, or he will be poisoned by food. For example, if there are 10 dishes, he can eat the food in the 2-nd, 3-rd and 4-th dishes but he can not eat the food in the 2-nd, 3-rd and 5-th dishes because there is the 4-th dish between them so it's not continuous. Furthermore, if he chooses to eat food in the i-th dish, he has to eat all food in that dish.

Bob's stomach has a strange feature that if there is at least t $(= 998857459)$ kg food in it, the weight in it will reduce t kg repeatedly until it is strictly lower than t kg. Also, if the weight of food in his stomach is exactly t kg, his stomach will be empty. Now Bob wants to eat the smallest number of dishes and remains no less than k kg food in his stomach. Can you tell him how many dishes he needs to choose?

Input

The first line contains two integers n and m $(1 \leq n \leq 100000, 1 \leq m \leq 10000)$, indicates the number of dishes and the number of queries.

The second line contains n integers $a_1, a_2, \cdots, a_n$, which is a permutation of $1, 2, \cdots, n$, indicates that there is $(a_i!)$ kg food in the i-th dish, where $s! = 1 \times 2 \times 3 \times \cdots \times s$.

The third line contains m integers, the i-th integer $k_i (1 \leq k_i < t)$ indicates a query with the lower bound of weight.

Output

Each line of m lines contains an integer indicates the number of dishes Bob needs to choose in that query. If there is no way to reach Bob's target, output "`-1`" instead.

Example

standard input	standard output
4 2 1 2 3 4 29 31	2 3

Problem H. Powers of Two

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Andrew has noticed that 7 and 9 are quite unusual digits. There are small powers of two that start with 1 in decimal notation ($2^0 = 1$), with 2 ($2^1 = 2$), ..., with 6 ($2^6 = 64$), and with 8 ($2^3 = 8$). However, the smallest power of 2 that starts with 7 in decimal notation is the enormous $2^{46} = 70,368,744,177,664$, and the smallest power of 2 that starts with 9 is even bigger: $2^{53} = 9,007,199,254,740,992$.

Having learned this, Andrew wants to study the powers of two a bit more. More precisely, he will generate a number p and present a positive integer k, and ask you: what is the k-th smallest power of two that starts with p in decimal notation?

Prepare for answering such requests! Note that it's possible to prove that there exist arbitrarily many powers of two that start with any given prefix.

Input

The input will contain one positive integer p with at most 30 digits without leading zeroes, and a positive integer k up to 70000.

Output

Output the k-th smallest non-negative integer r such that 2^r starts with p in decimal notation.

Examples

standard input	standard output
123 1	90
123 9	2226
123 12345	3510491
1234567890000000000 67891	192992076478480096905880

Note

To solve this problem, you may need the logarithm of 10 to the base of the mathematical constant e. Here it is in decimal notation with 503 decimal places after the decimal point:

$$
\begin{aligned}
\ln 10 \;=\; & 2. \\
& 30258509299404568401799145468436420760110148862877 \\
& 29760333279009675726096773524802359972050895982983 \\
& 41967784042286248633409525465082806756666287369098 \\
& 78168948290720832555468084379989482623319852839350 \\
& 53089653777326288461633662222876982198867465436674 \\
& 74404243274365155048934314939391479619404400222105 \\
& 10171417480036880840126470806855677432162283552201 \\
& 14804663715659121373450747856947683463616792101806 \\
& 44507064800027750268491674655058685693567342067058 \\
& 11364292245544057589257242082413146956890167589402 \\
& 567\cdots.
\end{aligned}
$$

Problem I. Resistance

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Recently, Sheep gets fascinated in a tower defense game called *Resistance*. But now he is stuck at the 99-th level. He doesn't know what to do and hopes for your help.

Resistance is a tower defense game. The map used in this game is an $n \times n$. A specific path consisting of several cells is given in which any two adjacent cells share a common edge. We guarantee that the path never passes by a cell twice. A village is locating at the destination of the path. Some evil invaders will land at the starting point, go along the path and attempt to invade the village.

To secure the village, Sheep has deployed k defense towers in the map to attack invaders. All defense towers are lying at some locations but neither on the path nor at the starting point or the destination. All defense towers can be divided into two kinds: **gun towers** and **bunkers**.

A gun tower is armed with the machine gun, whose effective range is up to r_1. Under normal circumstances, it selects the nearest target within the range and fire, causing d_1 damage points for its target. If there are more than one nearest enemies (with the same smallest Euclidean distance), the one which is closest to the village is the target. The rules of the game guarantee that there wouldn't be more ambiguities. After an attack, the machine gun will need t_1 frames of time for reloading ammunition, so it would not be used again until the $(x + t_1)$-th frame if it fired at the x-th frame.

A bunker is armed with a flame thrower, whose effective range is up to r_2. It can fire at every time without any restrictions like gun towers. It also selects a target satisfying the similar conditions as above. An attack will cause d_2 damage points for its target, and the attacked invader will start to burn in the next frame of time. At each frame when the invader is burning, it suffers from d_3 damage points. The fire will be quenched at the end of the $(x + t_2)$-th frame if was hit by a bunker at the x-th frame. Moreover, if a burning invader was hit by a bunker again, supposing that it was hit at the y-th frame, its burning time will be prolonged to the end of the $(x + t_2)$-th frame.

Now the round starts. There will be E invaders in total, the i-th of which will land at the starting point in the beginning of the i-th frame of time. After that, any invader will take a step forward along the path in each frame of time. All invaders can be also divided into two kinds: **privates** and **captains**.

A private is the most inferior warrior with no special ability whose initial health point is HP_1.

A captain is much more powerful whose initial health point is HP_2, with some level def of defense capability. Once attacked by a defense tower whose damage point is $damage$, he will suffer from $\max\{1, damage - def\}$ damage point. Note that attacks from different defense towers in any frame should be considered separately, but the damage caused by burning will not be considered as an attack within the defense capability.

An invader with health point less than zero is presumed dead and will be removed.

Let's make a summary of what will happen in a frame of time in this game.

- **The Enemy Step.** All alive invaders take a step forward along the path. Then, a new invader may land at the starting point.
- **The Fire Step.** All defense towers select their target. Then they fire together while those defense towers which have to target or wait for reloading ammunitions do nothing.
- **The Damage Estimation.** Calculate the damage points and deduce the health points of all invaders. The burning damage will also be calculated at this time.
- **The Ending Step.** All invaders with negative health points will be removed. If an alive invader is staying at the destination of the path, he invades the village successfully and will also be removed from the game.

Sheep has got some details about coming invaders, and he asks you to write a programme to simulate the game for the first T frames of time.

Input

The test data contains several test cases and the first line in input contains an integer $Tcase$ ($Tcase \leq 10$) representing the number of test cases.

For every test case, The first line contains five integers n, k, L, E and T indicating the size of the map, the number of the defense towers, the length of the path, the number of invaders and the number of frames among your simulation. We guarantee that $E \leq T$ in all cases.

The following L lines describe the path. The i-th of them contains two integers x_i and y_i describing the coordinate of the i-th cell in the path. We guarantee that any two adjacent cells in the path share a common edge. The first cell locating at (x_1, y_1) is the starting point and the last cell locating at (x_L, y_L) is the destination where the village is.

The next line contains seven integers $r_1, d_1, t_1, r_2, d_2, t_2$ and d_3 indicating all parameters about defense towers described as above. And then a line containing three integers HP_1, HP_2 and def describes parameters about invaders.

The next k lines describe all defense towers. The i-th of them contains three integers $type_i, x_i$ and y_i where (x_i, y_i) is the coordinate of the i-th defense tower, and if $type_i = 1$ the tower is a gun tower, or a bunker if $type_i = 2$.

The last line contains a string of length E, the i-th digit in which describes the type of the i-th invader where we use the character `1` to represent a private, and `2` to represent a captain.

All numbers in the input are positive integers and no more than 400.

Output

For the i-th test case, output "`Case #i:`" (without quotes) in the first line. Then output E lines describing the final states of all invaders.

The i-th line of them indicates the i-th invader:

- If it will invade the village successfully, output "`Arrive with Y HP(s).`" (without quotes) where `Y` represents his remaining HP.
- If it will be killed in the t-th frame, output "`Be killed in the t-th frame at (x,y).`" (without quotes), where (x, y) describes the location of his final resting place.
- If this invader will live up to the T-th frame but not arrive at the village, output "`Be alive at (x,y) with Y HP(s).`" (without quotes), where (x, y) represents his final coordinate in the T-th frame and `Y` represents his remaining HP.

Examples

standard input	standard output
2 5 1 5 3 8 2 1 2 2 2 3 2 4 2 5 2 5 1 2 5 1 1 8 8 2 1 1 3 111 5 1 5 3 3 2 1 2 2 2 3 2 4 2 5 2 5 1 2 5 1 1 8 8 2 1 1 3 111	Case #1: Be killed in the 3-th frame at (2,3). Arrive with 3 HP(s). Be killed in the 6-th frame at (2,4). Case #2: Be killed in the 3-th frame at (2,3). Be alive at (2,2) with 8 HP(s). Be alive at (2,1) with 8 HP(s).
2 5 1 5 3 8 2 1 2 2 2 3 2 4 2 5 2 5 1 2 5 1 1 8 8 2 1 1 3 211 5 1 5 3 8 2 1 2 2 2 3 2 4 2 5 2 5 1 2 5 1 1 8 8 2 2 1 3 211	Case #1: Arrive with 2 HP(s). Arrive with 3 HP(s). Be killed in the 6-th frame at (2,4). Case #2: Arrive with 0 HP(s). Arrive with 2 HP(s). Be killed in the 6-th frame at (2,4).

Problem J. Summon

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

LKMCFJ wants to cast a spell to summon the dragon. To do so, she needs to place n gems on the ground to form a **circular** summoning ring. She has prepared 4 kinds of different gems: the wind, the fire, the thunder and the lightning. Each of them is unlimited.

Some patterns of gems could explode and LKMCFJ would die if she does not avoid the risk. More precisely, some combinations of 4 continuous gems in the clockwise direction in given patterns are not allowed.

Now she wants to know how many different summoning rings she can make based on such limitations. Note that two necklaces are regarded as the same if they are the same after shifting one of them.

Since the answer can be very large, you only need to output the answer modulo 998244353.

Input

The first line of the input contains two integers n ($4 \leq n \leq 10^5$) and m ($0 \leq m \leq 256$) indicating the number of gems that are asked for the summoning ring, and the number of patterns of length 4 that could explode.

Each of the following m lines contains 4 integers describing a pattern of 4 continuous gems in the clockwise order that should be avoided. Here, we use integers $0, 1, 2$ and 3 to represent four different types of gems respectively.

Output

Output an integer in a line indicating the answer modulo 998244353.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>4 3
0 1 2 3
1 2 3 1
2 1 3 0</pre></td><td><pre>67</pre></td></tr>
</table>

Problem K. Tree

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Consider a rooted tree consisting of n vertices where the first vertex is the root. The i-th vertex is endowed with the value v_i.

Now you are given a positive integer k. An ordered pair of vertices (x, y), indicating the x-th and the y-th vertices in the tree, is valid if it satisfies the following conditions:

- $x \neq y$;
- The x-th vertex is not an ancestor of the y-th one;
- The y-th vertex is not an ancestor of the x-th one;
- The length of the shortest path connection these two vertices is up to k;
- Take the value of the lowest common ancestor of these two vertices, denoted by v_z. Then $v_x + v_y = 2v_z$.

Can you calculate the total number of valid pairs of vertices in this tree?

Input

The first line contains two integers n and k $(1 \leq n, k \leq 10^5)$ which is described as above.

The second line contains n integers, the i-th of which is the value of the i-th vertex v_i $(0 \leq v_i \leq n)$.

The third line contains $n - 1$ integers, the i-th of which represents the index of the parent vertex of the $(i + 1)$-th vertex.

We guarantee that the given tree is indeed a tree.

Output

Output the number of valid pairs of vertices in a line.

Two pairs (x_1, y_1) and (x_2, y_2) are regarded as the same if $x_1 = x_2$ and $y_1 = y_2$.

Examples

standard input	standard output
3 2 1 2 3 1 1	0
3 1 2 1 3 1 1	0
3 2 2 1 3 1 1	2

Problem L. Who is the Champion

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	256 megabytes

Soccer is probably the sport with the greatest fans in this country. However, the national team has a poor record on the world stage. In order to energize youth players, the Soccer National Championship is organized every year.

In this year, there are n teams taking part in the championship. As usual, the single round-robin system is adopted so that each team plays every other team once. A standard match askes ten players and a goalkeeper for both teams, which contains 22 players in total. They confront each other, defend their nets and attack the opponent's goal in a rectangular grass court.

In a 90-minute game, they try to shoot again and again. When the game is over, the team with higher goals will win and obtain 3 points in the scoreboard, while the loser will obtain nothing. A very special but common case is that both teams get the same goals, which is a draw; then both of them will obtain one point in the scoreboard.

At the end of the season, the league will declare the champion which is the team with the highest score in total. If two or more teams have the same highest score in total, the one with the highest goal differential (which is calculated as the number of goals scored in all league matches minus the number of goals conceded) will be the champion. The worst situation that several teams have the same highest score in total and the same goal differential will be solved by extra play-offs.

Input

The first line contains an integer n ($1 \le n \le 100$) indicating the number of teams.

Each of the following n lines contains n integers. The j-th integer in the i-th line, which is non-negative and up to five, represents the number of goals scored by the i-th team against the j-th team.

We guarantee that $a_{i,i} = 0$ for all $1 \le i \le n$.

Output

If the league can declare the champion, output the index of the team who will be the champion or output "`play-offs`" (without quotes), if extra play-offs will be organized.

Examples

standard input	standard output
2 0 1 2 0	2
2 0 1 1 0	play-offs
3 0 1 3 1 0 4 0 0 0	2

Problem M. XOR Sum

Input file: standard input
Output file: standard output
Memory limit: 256 megabytes

The bitwise exclusive-or sum of a sequence $a_1, a_2, \cdots, a_n$ is defined as $a_1 \oplus a_2 \oplus \cdots \oplus a_n$ where $\oplus$ represents the bitwise exclusive-or.

Now you are asked to calculate the value of

$$\sum_{k=1}^{t} \sum_{i=x}^{y} f(i,k)$$

where $f(i,k)$ is the bitwise exclusive-or sum of the sequence $1, 2, 3, \cdots, i^k - 2, i^k - 1, i^k$.

Since the answer can be very large, you only need to output the answer modulo $(10^9 + 7)$.

Input

The only line contains three integers t $(1 \le t \le 100000), x$ and y $(1 \le x \le y \le 10^{18})$ described as above.

Output

Output an integer in a line indicating the answer modulo $(10^9 + 7)$.

Examples

standard input	standard output
1 2 7	15
2 2 7	74
3 2 7	363
7 2 7	358035