CCPC 2019 Xiamen

October 20, 2019

A. Zayin and Bus

Zayin wants to go to the zoo with his friends. But his own car is not big enough to take so many people at the same time. So, he is going to use his own bus.

There are n seats (numbered from 1 to n) and $n-1$ passageways in the bus. Each passageway connects two seats. A person can walk from a seat to any other seats through one or more passageways. The entrance of the bus is connected to seat 1. There are n people, numbered from 1 to n. They get on the bus in the order from the first person to the nth person. Zayin needs to arrange seats for all n people. Everyone will follow the shortest path to his seat and then **immediately** start to sit down. The ith person gets on the bus and arrives at seat 1 at ith second. During each second, each person can move through a passageway. The ith person needs $a[i]$ seconds to sit down. After that nobody could arrived at the seat. (Refer to **Explanation** for more details.)

If everyone arrives at his own seat and sits well, then Zayin can start his bus. Zayin needs to find an arrangement that he can start his bus as soon as possible. Zayin wonders about the minimum possible time to finish the process.

Explanation

Assumed that $a[1] = 1$ and his destination is seat 1. He gets on the bus at $1st$ second and will finish sitting at $2nd$ second. Other people can still get on the bus at $2nd$ second, but $3rd$ second is not allowed because seat 1 has been blocked.

If his destination is seat 2, he will arrive at $2nd$ second and finish at $3rd$ second. Then nobody can reach seat 2 since $4th$ second.

Input

The first line of input contains an integer $T (1 \le T \le 15)$, denoting the number of test cases.

Each test case starts with a positive integer $n (1 \le n \le 10^5)$.

The next line contains $n-1$ integers. The ith number is $f[i]$, which means that there is a passageway between seat $f[i]$ and seat $i+1$. It is guaranteed that $1 \le f[i] \le i$.

The next line contains n integers. The ith number is $a[i]$ $(1 \le a[i] \le 10^8)$.

Output

For each test case, print the minimum possible time in a line.

Sample

Input	Output
2 3 1 1 1 3 2 3 1 2 1 3 2	6 6

B. Zayin and Elements

Zayin has n elements (numbered from 1 to n) and m props (numbered from 1 to m). Each prop can remove some specific elements. Prop i has shield a_i, base HP b_i, and bonus HP c_i. When Zayin wants to remove an element, he should obey the following rules:

Zayin should choose an alive prop (e.g. prop i) which can be used to remove this element. (Prop i is alive when the sum of its shield point, base HP and bonus HP is strictly greater than 0. That is to say, $a_i + b_i + c_i > 0$.)

1. If the shield (a_i) is greater than 0, the shield will be subtracted by 1, then go to 4, else go to 2.
2. If the base HP (b_i) is greater than 0, the base HP will be subtracted by 0.5, then go to 4, else go to 3.
3. The bonus HP (c_i) will be subtracted by 1, then go to 4.
4. Remove the element, and the process is finished.

Now Zayin wants to remove all the elements and maximized $\sum_{i=1}^{m} \lfloor b_i + c_i \rfloor$ ($\lfloor x \rfloor$ means the floor of x, the biggest integer which is not greater than x), but he is too busy so it is time for you to solve this problem.

Input

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

For each test case, the first line contains an integer $n (1 \le n \le 100)$, the number of elements.

The second line contains an integer $m (1 \le m \le 20)$, the number of props.

In the next m lines, the ith line firstly comes three integers a_i, b_i, c_i $(a_i, b_i, c_i \ge 0,\ a_i + b_i + c_i \le 10)$, representing the shield, base HP and bonus HP of prop i. Then follows a number t_i, indicates the number of elements can be removed by this prop. After that comes t_i numbers, representing the element numbers.

Output

For each test case, output the maximal $\sum_{i=1}^{m} \lfloor b_i + c_i \rfloor$. (or "$-1$" if there is no way to remove all the elements)

Sample

Input	Output
2 5 2 2 3 1 2 3 1 1 1 1 3 4 5 2 5 2 2 3 1 1 3 1 0 1 2 1 2	5 -1

C. Zayin and Fireball

Zayin is practicing magic at home. His house is an infinite 2-dimension plane. He can summon a fireball which burns a circle area to the ground. To prevent the house from lying in ruins, his dad Nothingnuo has to cast shield magic. The shield can protect a circle area from being destroyed.

This game repeat for N times. Each time Zayin summons a fireball and Nothingnuo places a shield. After the exploding, this shield will no longer work in the future. The process leaves some (or no) area ruined. And it will never recover.

Nothing wonders about the union of the ruined area.

Zayin's i-th fireball destroyed a circle area. It has its center located at (X_i, Y_i) and has a radius R_i. Nothing's shields are described similarly. For more details, you can refer to **Input**.

In short: There are $2N$ circles. Which can be described as $\{X_1, Y_1, R_1\}, \{X_2, Y_2, R_2\} \dots \{X_N, Y_N, R_N\}$ and $\{x_1, y_1, r_1\}, \{x_2, y_2, r_2\} \dots \{x_N, y_N, r_N\}$. (X, Y) is the coordinate of center, and R is radius. And you need to calculate $AreaSize(\cup_1^N \{X_i, Y_i, R_i\} \setminus \{x_i, y_i, r_i\})$.

Input

The first line of input contains an integer $T(1 \le T \le 30)$, denoting the number of test cases.

Each test case starts with a positive integer $N(1 \le N \le 500)$.

The next N lines give the description of the circles. The i-th line contains 6 integers: $X_i, Y_i, R_i, x_i, y_i, r_i$. $(|X|, |Y|, |x|, |y| \le 1000, 0 \le R, r \le 200)$.

Notice: It is guaranteed that all $\{X_i, Y_i, R_i\}$ are different with each other. And all $\{x_i, y_i, r_i\}$ are different with each other. But there may exist i, j that $\{X_i, Y_i, R_i\}$ and $\{x_j, y_j, r_j\}$ are the same.

Output

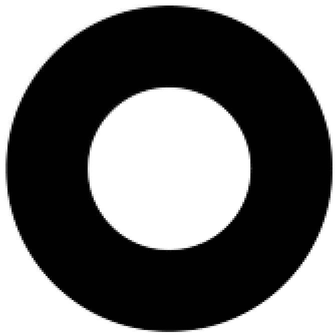
For each case print the ruined area.

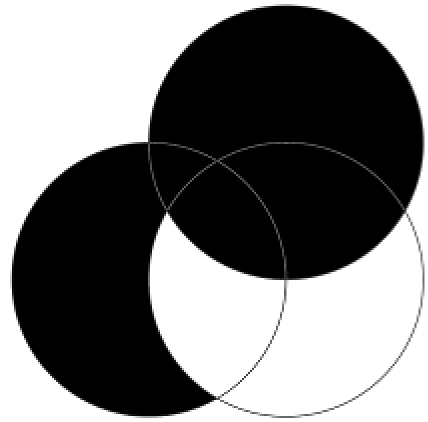
Absolute error up to 10^{-2} will be ignored.

Sample

Input	Output
2 1 0 0 2 0 0 1 2 0 0 2 2 0 2 2 2 2 233 0 51	9.424778 19.707963

Explanation





D. Zayin and Forest

We define $B(x)$ as the number of digit 1 in the binary representation of x. For example, $B(7) = B((111)_2) = 3$, $B(8) = B((1000)_2) = 1$, $B(9) = B((1001)_2) = 2$.

We define $F(x) = min\{\ y \mid (y > x) \wedge (B(y) \le B(x))\ \}$. For example, $F(4) = 8$, $F(5) = 6$, $F(6) = 8$.

Zayin has a forest (a set of trees), whose nodes are numbered from 1 to n. For each node (e.g. node x), if $F(x)$ is not greater than n, then the father of node x is $F(x)$, else, node x is the root of a tree.

We use $A[i]$ to denote the number of apples on the node i. Initially, all the $A[i] (1 \le i \le n)$ are equal to zero. In order to make his girlfriend happy, Zayin is going to perform magic on the tree. The magic consists of two types of operations:

1. Add x v : For every ancestor of node x (including itself), put v more apples on it. In other words, for every node (e.g. node y) on the path between node x and the root of its tree, let $A[y] = A[y] + v$.
2. Sum L R : Count the total number of apples on the nodes whose index is between L and R. In other words, you need to calculate $\sum_{i=L}^{R} A[i]$.

Input

The first line contains two integers n and m $(1 \le n \le 10^{18}, 1 \le m \le 10^5)$, where n is the number of nodes and m is the number of operations.

The next m lines describe the m operations in order. Each line contains three integers. The first of them is *op*. If $op = 1$, then the next two integers will be x and v, representing an **Add** operation. If $op = 2$, then the next two integers will be L and R, representing a **Sum** operation. $(1 \le x \le n,\ 1 \le v \le 10^9,\ 1 \le L \le R \le n)$

Output

For each **Sum** operation, output one line including one number, denoting the corresponding result.

Sample

Input	Output
8 6 1 1 1 1 3 2 1 5 3 1 7 4 2 1 5 2 4 8	10 23
1000000000000000000 2 1 1 1 2 1 1000000000000000000	60

E. Zayin and Camp

Nothingnuo is going to hold an icpc-camp. He has invited two super jbers, Zayin and dafeng, to come up with some sets of problems and asked them to give lectures at the same time. Due to the high price of these problems, Nothingnuo can just buy n problem sets from them. Each problem set is only used for a single contest. The n sets of problems have to be distributed into two divisions and two divisions can't share common sets of problems. That is to say, if divA uses x sets of problems to hold x contests, then divB will use the remaining $(n-x)$ problem sets to hold $(n-x)$ contests.

The students in the camp are divided into divA and divB. At the opening ceremony of the camp, people in two divisions are at the same level of energy at 0.

In each unit of time, each teacher has two choices of activities:

1. Give a lecture, after which students' level of energy will increase by 1.
2. Use a problem set that was distributed at first to organize a contest, after which students' level of energy will decrease by m.

Once the camp starts (at least one unit time after the opening), the level of students energy should keep larger than 0 constantly, or they will be too exhausted and will not take courses or testing anymore.

Zayin takes charge of divA and dafeng takes charge of divB, and they have so much technology to share that both of them want to extend their teaching time, but students in camp don't want to be so tired, for they are planning for tour after camp. So nothingnuo orders that the students' energy in divA after camp is exactly r, while divB's is exactly s.

Zayin wonders how many different schedules of the camp are there. We define the schedule of a particular division as the sequence of activities chronologically. And we define the schedule of a camp as the ordered pair of sequences of activities in two divisions. For two camping schedules, we regard them as identical if and only if divA has the same activity sequence in two schedules and divB does same as well (Two activities are same if they are both lecture or both contest. That is to say, we ignore the difference between the problem sets). For example:

Schedule 1: Div.A (Lecture, Lecture, Lecture, Testing), Div.B (Lecture, Testing, Lecture, Lecture).

Schedule 2: Div.A (Lecture, Lecture, Testing, Lecture), Div.B (Lecture, Testing, Lecture, Lecture).

These two schedules are different. But if we change the sequence in schedule 2 Div.A to (Lecture, Lecture, Lecture, Testing), then it will be the same as the first.

Input

The first line of input contains an integer $T(1 \le T \le 10^4)$, denoting the number of test cases.

Each test case contains one line with four integers n, m, r, s, where n is the number of problem sets, m is the physical strength cost of each problem set, r is the rest strength of divA, and s is the rest strength of divB after camp. ($1 \le n, m \le 10^7$, $n \times m \le 10^7$, $1 \le r, s \le 5 \times 10^6$)

Output

For each case, print an integer in a line, denoting the number of different camp schedules. Because the result may be too large, you just need to output the answer module 998244353.

Sample

Input	Output
2 3 2 1 2 4 1 1 1	55 42

Explanation

For the first case:

When div.A is distributed 0, 1, 2, 3 sets of prob, the number of schedules are 1, 1, 3, 12 respectively.

When div.B is distributed 3, 2, 1, 0 sets of prob, the number of schedules are 30, 7, 2, 1 respectively.

So the number of different schedules is $1 \times 30 + 1 \times 7 + 3 \times 2 + 12 \times 1 = 55$.

F. Zayin and Dirichlet

We define three types of **fundamental functions**,

$$\mu(n) = \begin{cases} (-1)^{\sigma} & , n \text{ has no square factor,} \\ & \quad \textit{and } \sigma \text{ is the number of factors of } n \\ 0 & , \text{otherwise} \end{cases}$$

$$1(n) = 1$$

$$id_k(n) = n^k \qquad , k = 1, 2, 3, \ldots$$

The Dirichlet product of functions f and g (represented as $f * g$) is a function h where

$$h(n) = \sum_{d|n} f(d) g(\frac{n}{d})$$

The Dirichlet product of more than two functions (for example, there are q functions $f_1, f_2, \cdots, f_q$) is defined as

$$f = (((f_1 * f_2) * f_3) * \cdots) * f_q$$

Zayin and Ziyin now have many **fundamental functions** (at least one). They calculate the Dirichlet product of them all and get the result f. Zayin finds that for any prime p, $f(p^c)$ is a polynomial of p of degree n, i.e. $f(p^c) = \sum_{i=0}^{n} a_i p^i$. But a_i is so large that he can only tell you a_i mod 998244353, i.e. x_i where $0 \le x_i < 998244353$ and $x_i \equiv a_i \pmod{998244353}$ holds.

And Ziyin finds that for any prime p, $f(p^d)$ is also a polynomial of p, but she doesn't want to tell you what it looks like. So it's your task to find out that polynomial.

If there are multiple solutions, please print the smallest one. Polynomial $P(x) = \sum_{i=0}^{n} a_i x^i$ is smaller than $Q(x) = \sum_{i=0}^{m} b_i x^i$ if and only if $n < m$, or $n = m$ and $\exists i \in [0, n]$, $(\forall j \in [i+1, n]\ a_j = b_j) \wedge (a_i < b_i)$ (after modulo 998244353).

If there is no solution, or the number of fundamental functions used by the smallest solution is more than 10^5, please print -1.

Input

The first line contains three integers n, c and d. ($0 \le n \le 1000$， $1 \le d \le c \le 100$)

The second line contains $n+1$ integers $x_0, x_1, \ldots, x_n$. ($0 \le x_i < 998244353$, $x_n \ne 0$)

Output

Print -1 in the only line if it satisfies the corresponding conditions above. Otherwise the output contains two lines.

The first line contains one integer m, representing the degree of Ziyin's polynomial.

The second line contains $m+1$ integers $b_0, b_1, \cdots, b_m$, representing $f(p^d) = \sum_{i=0}^{m} b_i p^i$. You must make sure $b_m \ne 0$ when $m > 0$. Due to the answers are so large, please print answers modulo 998244353.

Sample

Input	Output
2 2 1 1 1 1	1 1 1
2 2 1 998244352 998244352 1	-1

Explanation

In the first sample, $f(p^2) = p^2 + p + 1$. $f(n)$ can be regarded as the sum of divisors of n, so that $f(p) = p + 1$.

In the second sample, $f(p^2) = p^2 - p - 1$ can not be Dirichlet product of fundamental functions.

G. Zayin and Count

Zayin and Taotao are poor in mathematics. In order to improve their math skill, they count numbers together today. They write a number every second in the ascending order, such as $0, 1, 2, 3$, etc. However, they can't recognize all the Arabic letters. As a result, they both ignore numbers containing the Arabic letters they don't know. For example, if Zayin only knows 0 and 1, he will write down $0, 1, 10, 11$, etc.

For each Arabic letter, you know whether Zayin and Taotao can recognize it. Both Zayin and Taotao can recognize at least two Arabic letters. Besides, you know the number Zayin will write in x-th second. Can you write a program to calculate the number that Taotao will write at this time?

Input

The first line of input contains an integer $T (1 \leq T \leq 10^4)$, denoting the number of test cases.

For each test case, you will get a boolean array a which contains exactly 10 booleans in the first line. If $a_i = 1$ (the index starts at 0), Zayin can recognize Arabic letter i. Otherwise, Zayin can't recognize Arabic letter i.

Similarly, you will get a boolean array b which contains exactly 10 booleans in the second line. If $b_i = 1$ (the index starts at 0), Taotao can recognize Arabic letter i. If $b_i = 0$, Taotao can't recognize Arabic letter i.

In the third line, you will get the number Zayin will write in x-th ($x \leq 2^{64}$) second.

Output

For each test case, you should write down the number that Taotao will write in x-th second in a single line.

Sample

Input	Output
1 1 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 20	10

H. Zayin and Obstacles

Recently, Zayin became obsessed with a tower defense game called Arknights. The most special level is the 5th level of chapter 4: Don't panic. The most special thing about this level is that an enemy will continue taking radioactive damage after passing through the Active Originiums. As the most handsome man in the world, he tried to put obstacles in various positions, making all the enemies to be killed before reaching the end. After many attempts, he finally won the three stars clearance award.

Interestingly, that night Zayin dreamt that he had become the King Slayer (the leader of enemies). In the dream, the map of the tower defense game becomes a cube with edge length n. Zayin can take a step forward, backward, up, down, left or right from his current position every second, but can't get out of the map or walk through a grid of obstacles.

As in the original game, the player, who wants to kill all the enemies, can put obstacles many times. In particular, in the dream, every time players can place obstacles at all the coordinates (x, y, z) which satisfies $a \le x \le d$, $b \le y \le e$, $c \le z \le f$, and has no obstacles yet.

Now, Zayin knows that the player will place obstacles m times and the locations of each time. In order to avoid dying on the way, Zayin wants to go to the end as fast as possible. But unfortunately, Zayin got lost for so long that all the obstacles were placed. Now, Zayin asks you for help. Can you help him get out of the map as quickly as possible?

Input

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

Each test case starts with two integers n and m in a line, where n is the size of the map and m is the number of times the player puts obstacles. $(1 \le n \le 100, 1 \le m \le 1000)$

The next m lines contain six integers a, b, c, d, e, f in each line, denoting the range of obstacles the player puts. $(1 \le a \le d \le n, 1 \le b \le e \le n, 1 \le c \le f \le n)$

Then there are six integers $x_1, y_1, z_1, x_2, y_2, z_2$. The first three integers represent where Zayin is now, and the last three integers represent where the end point is.

Output

For each testcase, output an integer in a line, denoting how many seconds at least Zayin need to go to the end. If Zayin can't go to the end, output -1.

Sample

Input	Output
3 3 0 1 1 1 3 3 3 3 1 2 1 1 2 3 3 1 1 1 3 3 3 3 3 2 1 1 2 2 3 1 1 2 2 3 2 1 2 2 3 3 2 1 1 1 1 1 3	6 -1 14

I. Zayin and Coin Game

Zayin is fond of collecting coins. And Zayin has collected n different coins in total. Now he wants to play a game with these coins. He places all these coins on a desk in a **circle**, numbered from 1 to n in clockwise order. In one step Zayin can choose exactly k consecutive coins and flips them at the same time, which means that the heads of these coins which are facing up become facing down and which are facing down become facing up.

Now he wonders whether the original state can reach the final state in finite steps.

Input

The first line contains an integer $T (1 \le T \le 100)$, the number of test cases.

Then each test case contains 3 lines. The first line contains two integers, n and k $(1 \le k \le n \le 10^5, \sum n \le 10^6)$, whose meanings have been described above. The next two lines contain two strings, s and t, respectively $(|s| = |t| = n)$, which only contain the digits 0 and 1.

String s represents the initial state of the n coins. If the head of the i-th coin faces up, then the i-th character of s is 1. If the head of i-th coin faces down, the i-th character of s is 0. String t represents the desired final state of the n coins in the same way as s.

Output

If it is possible for Zayin to reach the state represented by t from the state represented by s in finite steps, outputs “Yes”; otherwise, output “No” (without the quotes).

Sample

Input	Output
4 3 2 000 100 3 2 000 110 6 3 100100 100101 8 3 00100100 00100101	No Yes No Yes

J. Zayin and Tree

Zayin gives you a tree with n nodes (numbered from 1 to n). For each node i, there is a value a_i.

You could choose a simple path on the tree. Suppose that P is the number of nodes on the path, R is the maximum value on the path, L is the minimum value on the path.

Please choose a path so that $P - R + L$ is minimal. Output the minimum value of $P - R + L$.

Input

The first line contains an integer T, the number of test cases.

For each test case, the first line contains an integer n, denoting the number of nodes.

The next line contains n integers, the i-th integer is a_i. $(0 \le a_i \le 10^6)$

The following $n-1$ lines describe the edges of the tree. Each line contains two integers u_i and v_i, denoting that there is an edge between node u_i and v_i.

The sum of n in all test cases won't exceed 10^6.

Output

For each test case, output the minimum value of $P - R + L$.

Sample

Input	Output
2 5 4 5 3 4 2 1 2 2 3 3 4 3 5 5 4 4 1 1 2 1 2 2 3 3 4 3 5	0 -1

Hint

In the first example, you can choose path (2,5).

In the second example, you can choose path (2,3).

K. Zayin and String

Zayin likes chatting online with his girlfriend and he has compiled a dictionary of love words. The dictionary contains m words named $w_1, w_2, \ldots, w_m$, each of which has its own love value $v_1, v_2, \ldots, v_m$.

All words are strings consisting of only lowercase English letters. And the love value of any word that isn't in the dictionary is 0.

By using this dictionary, Zayin can define the sweetness of a string S.

$$sweetness(S) = \frac{\sum_{l=1}^{|S|} \sum_{r=l}^{|S|} love(S(l,r))}{|S|}$$

Where $S(l,r)$ means the substring of S which starts with l-th character and ends with r-th character (both inclusive) of S, $|S|$ means the length of string S, and $love(s)$ means the love value of string s, as defined above.

This means the sweetness of a string S is the sum of love value of all its substring devided by the length of S.

Now Zayin's girlfriend has sent Zayin a message consisting of a string S and asks Zayin a question, what is the maximum sweetness of all subsequences of S? A subsequence of S is a string that can be derived from string S by deleting some or no elements without changing the order of the remaining elements.

When Zayin is dating with his girlfriend, his IQ drops from 301 to 31. So he asks you to help him solve the problem.

Input

The first line of input contains an integer $T (T \leq 80)$, indicating the number of cases.

For each case, the first line consists of two numbers $n (1 \leq n \leq 1000), m (1 \leq m \leq 1000)$, the length of string S and the size of dictionary. The second line is a string S followed by m lines, each line contains a string w_i and an integer v_i.

We guarantee that for each case, $n, m < 1000, \sum |w_i| \leq 4000, 0 \leq v_i \leq 10^5$ and w_i are pairwise different.

For each input file, $\sum n \leq 3000, \sum m < 2000, \sum |w_i| \leq 10000$.

All strings consist of only lowercase English letters.

Output

For each test case, you need to output the maximum sweetness of all subsequences. In order to avoid output floating point number, you need to output the answer modulus 998244353. If the answer is $\frac{A}{B}$, you need to output the value of $A \times B^{-1} \mod 998244353$.

Sample

Input	Output
3 17 3 woyaoakccpcxiamen ak 5 ccpc 6 xiamen 8 33 3 niweishenmezhengtianxunlianbuliwo wo 3 se 1 zayin 7 39 2 programmingcontestandmewhichisimportant me 11 gg 2	499122179 499122178 499122182

Hint

For test case 1, the subsequence with maximum sweetness is “ak”;

For test case 2, the subsequence with maximum sweetness is “wo”;

For test case 3, the subsequence with maximum sweetness is “me”.

L. Zayin and Raffle

Zayin is playing an interesting computer game Battlegrounds, which is developed by Tendollars Company. In this game, there are m weapons that players can use, such as AK-47 Assault rifle, shotgun, Rocket Propelled Grenade and so on. Those m weapons are numbered from 1 to m.

Players can start a match. In a match, up to one hundred players equipped with weapons parachute onto an island and fight with each other. The last player to survive will win the game.

For the same weapon, the player can be equipped with at most one. So we can learn the strength of a player from the set of weapons he owns.

Firstly, I want to show you how to represent a set of weapons using binary number. We can use an m-bit binary number x to denote a set S. The i-th bit (from least significant to most significant) of x represents whether S contains the i-th weapon. When the i-th bit is 1, the set contains the i-th weapon. Otherwise, the set does not contain it. We call the number x is the **numberic representation** of the set S. For simplicity, we just use the decimal form of x to describe a set S in the rest of this problem description.

For example, when $x = 6$, its binary form is 110, which means weapons 2 and 3 are in the set.

In the beginning, Zayin doesn't have any weapons (the set of his weapons can be denoted as 0). So he is very easy to be defeated by other players. Zayin needs to collect more weapons to become stronger.

He is going to participate in an n-round raffle. In each round, Tendollars Company will show k boxes, each of which contains some types of weapons. Zayin will get **exactly one box** in this round and obtain all weapons in this box. The probability that Zayin gets the j-th box is $p_j(1 \le j \le k)$. Because programmers in the company wrote some bad codes, these probabilities $p_j(1 \le j \le k)$ wouldn't change among rounds. The set of weapons in the j-th box in i-th round is a_{ij}.

Let $f(x)$ be the probability that the final set of weapons Zayin obtains is x.

For all possible sets $x(0 \le x \le 2^m - 1)$, please calculate the value of $f(x)$, modulus 1000000007.

Input

The first line contains three integers, $n, m, k(1 \le n \le 100000, 1 \le m \le 16, 1 \le k \le 8)$, the number of rounds of raffle, the number of different weapons, the number of boxes in each round.

The second line contains k intergers, $b_1, b_2, ..., b_k(0 \le b_j \le 1000000000)$. The probability that Zayin gets j-th box in a round is $p_j = \frac{b_j}{1000000000}$. It is guaranteed that the sum of all b_j is 1000000000.

Following n lines describe weapon sets in each box in each round. The i-th line contains k integers, $a_{i1}, a_{i2}, ..., a_{ik}$. a_{ij} is the numberic representation of set of weapons in the j-th box of i-th round.

Output

Output 2^m lines, the x-th line contains the value of $f(x-1)$ (modulus 1000000007), $x = 1, 2, \ldots, 2^m$

Sample

Input	Output
2 2 2 400000000 600000000 1 0 2 1	0 200000002 680000005 120000001

Hint

In the sample above, the probabilities to get the two boxes are $\frac{2}{5}$ and $\frac{3}{5}$ respectively.

The probablities of sets after each round are shown in the following table.

set of weapons	00	01	10	11
Initial	1	0	0	0
After 1st round	$\frac{3}{5}$	$\frac{2}{5}$	0	0
After 2nd round	0	$\frac{3}{5}$	$\frac{6}{25}$	$\frac{4}{25}$