2020 SYSU Collegiate Programming Contest, Onsite

November 29, 2020

A. Undercover

There are n gamers and a host playing a game "Who is Undercover". The gamers are numbered from 1 to n. The host randomly selects exactly 1 gamer as undercover, but he won't tell this to others. The host will ask each gamer "Who is undercover?". Each gamer can either answer "Undercover is number a_i", or "Undercover isn't number a_i". Also, gamer can say so about himself ($a_i = i$).

After that, the host will tell gamers that exactly m answers are truths, and the other answers are lies. Now you are a spectator, can you identify who told the truth, and who told the lie?

Input

Input consists multiple test cases, please proceed to end of file (EOF).

For each test case, the first line contains two integers n and m ($1 \le n \le 10^5, 0 \le m \le n$), denoting the total number of gamers, and the number of gamers who told the truth. The next n lines contain the gamers' answers. The i-th line of them contains either "$+a_i$" (without the quotes), if the i-th gamer says that the undercover is number a_i, or "$-a_i$" (without the quotes), if the i-th gamer says that the undercover isn't number a_i (a_i is an integer, $1 \le ai \le n$).

It is guaranteed that at least one gamer exists, such that if he is the undercover, then exactly m people told the truth.

The sum of n will not exceed 10^6.

Output

For each test case, output a test case number (see the sample below), and then print n lines. The i-th line of them should contain "Truth" if the i-th gamer has told the truth for sure. Print "Lie" if the i-th gamer lied for sure, and print "Not defined" if he could lie and could tell the truth, too, depending on who is the undercover.

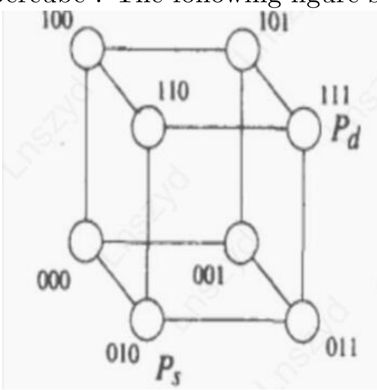
Sample

Input	Output
1 1	Case #1:
+1	Truth
3 2	Case #2:
-1	Not defined
-2	Not defined
-3	Not defined
4 1	Case #3:
+2	Lie
-3	Not defined
+4	Lie
-1	Not defined

B. Orienteering

"Orienteering" is a popular sporting event among young people. The participants must rely on the map marked with a number of check points and direction lines, and with the help of the compass, choose their own routes, look for each checkpoint in turn, and those who finish the competition in the shortest time is the winner.

Since Emily plays quite well in the orienteering games, her friends decided to confuse her in the following way. There are 2^n points indexed from 0 to $2^n - 1$, and there is an undirected path between two points u, v if and only if only one bit of their binary representations is different. In other words, points are connected like "hypercube". The following figure shows the situation when $n = 3$.



Now Emily is going to pass all the checkpoints in turn, so she wants to know the shortest path between two given checkpoints.

Input

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

For each test case, there is only one line with three integers n, x, y $(1 \le n \le 20,\ \ 0 \le x, y < 2^n)$.

Output

For each test case, output a line containing the answer path separated by a single space.

If there are multiple solutions, any solution will be accepted.

Sample

Input	Output
3 1 0 0 2 1 0 3 2 7	0 1 0 2 6 7

C. Compare

Give you two decimal real numbers x and y, you should compare x and y. That is to say, you need to determine that x is greater than y, equal to y or less than y.

Input

The first line of input contains an integer T $(1 \le T \le 50)$, denoting the number of test cases.

Each test case contains two lines. The first line contains a decimal real number x, and the second line contains a decimal real number y.

It's guaranteed that $3 < x, y < 4$, and there are at most 100 digits after decimal point, and there are no leading zeros.

Output

For each test case, output a line. If $x > y$, print “$>$”. If $x = y$, print “$=$”. If $x < y$, print “$<$”.

Sample

Input	Output
3 3.14 3.13 3.14 3.14000 3.1415926535897932384626 3.1415926535897932384627	$>$ $=$ $<$

D. Decode

You must have heard that in a computer, characters are stored in ASCII code. ASCII code can represent 128 symbols, including 52 English letters, 10 Arabic numerals, 33 punctuations, and some control characters. When store in computer, every ASCII code take exactly 1 byte (8 bits). These 128 symbols are enough to represent English sentences, but if we want to denote some Chinese characters, ASCII code is not enough.

To deal with such situation, Unicode is proposed. Unicode have up to 2^{16} symbols, from 0x0000 to 0xFFFF (0x means they are hexadecimal numbers), exactly 16 bits, so we can simply use 2 bytes to store it. Though Unicode can encode more characters, in some situation it will cause waste of storage space. To store an article in pure English, if we use ASCII code, every character takes only 1 bit. But if we use Unicode, it will take double space.

To reduce such waste, computer uses a variable length encoding to store the Unicode, which is called UTF-8 encoding:

Unicode	UTF-8	Symbols
0x0000 − 0x007F	0XXX XXXX	128
0x0080 − 0x07FF	110X XXXX 10XX XXXX	2048
0x0800 − 0xFFFF	1110 XXXX 10XX XXXX 10XX XXXX	65536

X means the available binary bits that can be used to encode. For example, if we want to encode "0x003A" which is in range "0x0000 − 0x007F", we encode it with first pattern: "0XXX XXXX", write it to a 7-bit binary representation: 011 1010, and fill the 'X' position from left to right, get 0011 1010.

When we decode UTF-8 to Unicode, first determine which format it belongs to, then take out all 'X' position bits in that format, and concatenate them in sequence into a long binary number, then change this number to 8-bit hexadecimal format, finally get the corresponding Unicode sequence.

For example, when we receive 1100 1010 1011 1100 0111 1010 0100 0000, first we can determine it is a second format followed by two first formats:

1100 1010 1011 1100 (second format) || 0111 1010 (first format) || 0100 0000 (first format)

Take out the available bits in each format we get:

0 1010 11 1100 || 111 1010 || 100 0000

Concatenate them in sequence and get the long binary number:

1010111100 || 1111010 || 1000000
700 122 64

Then write it to 8-bit hexadecimal format, we get the Unicode sequence 02BC007A0040.

In this problem, you need to implement a UTF-8 decoder (do the things showed above), which will receive a UTF-8 encoded string, and output a corresponding Unicode sequence.

Input

The first line contains one integer t, representing the number of test cases. ($t \le 10$)

In the next t lines, each line contains one string with only '0' and '1', representing the UTF-8 sequence. It's guaranteed that it is a valid UTF-8 sequence.

The total length of these t strings won't exceed 2×10^5.

Output

Output t lines, and each line contains one string, representing the corresponding Unicode sequence.

Sample

Input	Output
2 11001010101111000111101001000000 110010101011110001111010111011111001001010111101	02BC007A0040 02BC007AF4BD

E. Reading

Zayinese is a language formed by Zayinese words. Zayinese words are sequences containing 26 lowercase Latin letters ('a' to 'z').

Now Kaiqi is learning Zayinese. He has already mastered m Zayinese words and is going to read n Zayinese books. The n books are numbered from 1 to n.

He uses the rules below to decide the order to read these books.

1. Kaiqi picks a book which contains the least number of unknown words and then reads it first.

2. After reading the first book, Kaiqi masters all words from it. And then he picks a book which contains the least number of unknown words for him now and reads it.

3. After reading the second book, Kaiqi masters all words from it. And then he picks a book which contains the least number of unknown words for him now and reads it.

...

We will show you the words mastered by Kaiqi initially and the words in each book. Please output the order of books Kaiqi is going to read. At every moment that two books have the same least number of unknown words, Kaiqi picks the book which occurs first, in other words, has the smaller number.

Input

First line consists of an integer T, denoting the number of test cases.

For each test case, the first line consists of two integers, m and n. ($m, n \leq 100000$) The second line consists of m strings, the words mastered by Kaiqi initially. In the next n lines, each line consists of an integer C_i, followed by C_i strings, the words in the i-th book.

Each word contains only lowercase Latin letters ('a' to 'z') and its length is between 1 and 10 (both inclusive). The total number of words in a test file is no more than 300000.

Output

For each test case, output n integers in a line, which means the order of books to read.

Sample

Input	Output
2 1 4 apple 2 banana cat 2 apple banana 3 cat dog egg 1 fly 2 2 cat dog 1 dog 1 cat	2 1 4 3 1 2

Explanation

Here is the explanation for the first test case.

Initially, Kaiqi only knows one word 'apple', so the number of unknown words that the four books contain is 2,1,3,1, respectively. Book 2 and Book 4 have the least number of unknown words, Kaiqi picks Book 2 to read.

After reading Book 2, Kaiqi knows 2 words 'apple' and 'banana', so the number of unknown words that Book 1,3,4 contain is 1,3,1, respectively. Book 1 and Book 4 have the least number of unknown words, Kaiqi picks Book 1 to read.

After reading Book 1, Kaiqi knows 3 words 'apple', 'banana' and 'cat', so the number of unknown words that Book 3,4 contain is 2,1, respectively. Book 4 have the least number of unknown words, Kaiqi picks Book 4 to read.

Then there is only one book left. So Kaiqi reads Book 3.

F. Competition

In the programming contest, there is a very interesting phenomenon, the participants are more likely to try to solve problems that have been already solved by other teams. And the more teams have passed the problem, the easier this problem will be considered, though in some cases this is not the case.

When a hard problem is solved very early by a strong team, it will lead much more teams try to solve it, and with more and more teams passed this problem, almost all the participants will think it must be very easy, and try their best to solve it, though it is actually hard. Sometimes, the teams will be greatly inspired by such idea, and finally they will solve the problem which was beyond their abilities.

This problem is about it. There are n teams in one contest, each team has an "ability value" a_i , and we also define a problem's "difficult value" $Diff_j$, when $a_i \geq Diff_j$, we think team i can solve problem j. In these n teams, some teams are familiar with each other and some are not. For some team pair (a, b), we can define their "inspire value" $Ins_{a,b}$, which means if one of these two teams passed a problem, the other team's ability to **this** problem will be added by $Ins_{a,b}$, so they will have more chance to solve **this** problem. And there are m problems, for every problem, we want you to determine how many teams will pass this problem. (Note, if at any moment, one team's ability value to a problem become not less than the difficult value of this problem, the team will certainly pass this problem)

Input

First line contains one integer t, representing the number of test cases. ($t \leq 5$)

For each test case:

The first line contains three integers n, m, K, representing the number of teams, the number of problems and the number of team pairs which are familiar with each other. ($1 \leq n, m, K \leq 10^5, K \leq \frac{n(n-1)}{2}$)

The second line contains n integers a_i, represent every team's "ability value" ($0 \leq a_i \leq 10^5$)

The third line contains m integers $Diff_i$, represent every problem's "difficult value" ($1 \leq Diff_i \leq 10^6$)

Next K lines describe the familiar team pairs. Every line contains three integer $a, b, Ins_{a,b}$, represent "inspire value" of team pair (a, b). ($1 \leq a, b \leq n, a \neq b, 1 \leq Ins_{a,b} \leq 10^4$, pair (a, b) will only appear once)

Output

For each test case, output m integers in one line, represent how many teams will pass the problem.

Sample

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

Explanation

When difficult value is 5, only team number 3 can pass this problem by themselves, and that will inspire team 1 and team 2, their ability values will become 5 and 2. Then team 2 will pass this problem, that will inspire team 1, let its ability value become 4, but is still not enough to solve the problem with difficult value 5. So, finally only two team pass this problem.

If the problem's difficult value is 4, at last team 1's ability value is enough, so all the three teams pass the problem.

G. Sum

Give you $M, A = (a_1, a_2, \ldots, a_{N-1}, a_N), B = (b_1, b_2, \ldots, b_{N-1}, b_N)$, and you are required to find the number of X that satisfies $X = (x_1, x_2, \ldots, x_{N-1}, x_N), a_i \le x_i \le b_i, M = \sum_{i=1}^{N} x_i$.

Input

The first line contains two intergers N, M.

The second line contains N integers $a_1, a_2, \ldots, a_{N-1}, a_N$.

The third line contains N integers $b_1, b_2, \ldots, b_{N-1}, b_N$.

$1 \le N \le 8,\ |M| \le 1600,\ |a_i| \le 200,\ |b_i| \le 200$

Output

Output a line with an integer, denoting your answer.

Sample

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

H. Sequence

Give you a sequence with length N. You need to change some numbers in it, and then divide it into M consecutive subsequences, such that every consecutive subsequence contains all the numbers from 1 to K. How many numbers at least do you need to change?

Input

The first line contains three integers N, M, K.

The second line contains N integers A_i, denoting the sequence.

$1 \le N, K \le 200000$, $1 \le M \le 10$, $1 \le A_i \le K$

Output

Output one line with one integer, the minimum number of numbers to change. If there is no solution, print -1.

Sample

Input	Output
10 3 3 1 3 3 2 1 3 1 1 2 2	2

Explanation

Change the third number to 2, and change the 10-th number to 3.

And then devide it into $\{1,3,2\},\{2,1,3\},\{1,1,2,3\}$

I. TianHe-IIA

TianHe-II is located in the beautiful East Campus of Sun Yat-Sen University, and has been the fastest supercomputer in the world from June 2013 to June 2016. In September 2017, National Supercomputer Center in Guangzhou announced to upgrade TianHe-II supercomputing system by the end of the year, replacing the original Intel Xeon Phi accelerator with the domestic accelerator matrix 2000. The upgraded TianHe-II is called TianHe-IIA. The number of nodes has increased from 16000 to 17792, and the floating-point performance has increased from 54.9pflops to 94.97pflops.

During a TianHe-IIA's schedule, the cluster workload manager allocates n computing nodes to users so they can perform work. To simplify this problem, all the n computing nodes are considered to be in a line from left to right and indexed from 1 to n. At the beginning each node i holds a nonnegative integer a_i. Then the nodes are ready to work, by executing any of the four types of operations:

1. $1\,l\,r$: $\forall i \in [l, r], a_i \to \phi(a_i)$
2. $2\,l\,r\,x$: $\forall i \in [l, r], a_i \to x$
3. $3\,l\,r\,x\,y$: $\forall i \in [l, r], a_i \to \left(\sum_{i=l}^{r} a_i^x\right) \mod y$
4. $4\,l\,r\,x\,y$: Print $\left(\sum_{i=l}^{r} a_i^x\right) \mod y$

Now WuK wants you to simulate the operation of TianHe-IIA.

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, there is only one line with four integers n, m, $seed$, $vmax$ $(1 \le n, m \le 10^5$, $1 \le seed, vmax \le 10^7)$. The initial values and operations are generated using following pseudo code:

```
def rnd():
    ret = seed
    seed = (seed * 7 + 13) mod 1000000007
    return ret
for i = 1 to n:
    a[i] = (rnd() mod vmax) + 1
for i = 1 to m:
    op = (rnd() mod 4) + 1
    l = (rnd() mod n) + 1
    r = (rnd() mod n) + 1
    if (l > r):
        swap(l, r)
    if (op == 2)
        x = (rnd() mod vmax) + 1
    if (op == 3) or (op == 4):
        x = (rnd() mod vmax) + 1
        y = (rnd() mod vmax) + 1
```

Output

For each operation of type 4, output a line containing the answer.

Sample

Input	Output
1 7 7 7 7	4

Explanation

$\phi(i) = \sum_{j=1}^{i-1}[\gcd(i,j) = 1]$. Moreover, $\phi(0) = 0, \phi(1) = 1, \phi(2) = 1, \phi(3) = 2, \phi(4) = 2, \ldots$

$0^0 = 1, 0^1 = 0, 0^2 = 0, 0^3 = 0, \ldots$

The following table explains the sample in detail.

a_i	operation
1 7 7 7 7 7 7	3 7 7 2 4
1 7 7 7 7 7 2	2 1 6 7
7 7 7 7 7 7 2	3 1 2 3 5
1 1 7 7 7 7 2	1 1 5
1 1 6 6 6 7 2	2 3 7 7
1 1 7 7 7 7 2	1 1 5
1 1 6 6 6 7 2	4 1 4 3 5

J. Traffic Lights

There are N traffic lights from WKQ's home to his company. The cycles of all the traffic lights are C.

For the i-th traffic light, from time 0 to time A_i, it will be green, and then from time A_i to time C, it will be red, and then this process repeats. That is to say, if you arrive at the traffic light in $[0, A_i)$, you can pass it without waiting, but if you arrive in $[A_i, C)$, you need to wait until time C.

Now you know that the time WKQ needs to go from his home to the first traffic light is D_0, the time from the i-th traffic light to the $i+1$-th traffic light is D_i, and it takes WKQ D_N to go from the last traffic light to his company.

Now there are Q queries that WKQ wonders if he starts to go to his company from his home at time t, how much time he will spent on the road?

Input

The first line contains three integers N, C, Q, whose meanings have been mentioned above.

The second line contains $N+1$ numbers D_i (from D_0 to D_N).

In the next Q lines, each line contains an integer t, denoting the time WKQ starts to go.

$1 \le N, Q \le 100000,\ \ 1 \le C, D_i \le 10^9,\ \ 0 \le t \le 10^9,\ \ 1 \le A_i < C$

Output

Print Q lines, and each line contains an integer, denoting the time WKQ needs to go to his company.

Sample

Input	Output
3 5 5 3 4 2 1 2 2 1 0 1 2 3 4	6 10 9 8 7

K. Cards

Mingming and Baibai are good friends. Mingming is indulged in a computer game and his target is to **collect two sets of cards**. A set of cards consists of n different cards, numbered from 1 to n.

After a game ends, a card will appear and Mingming will get the card. The probability that the i-th card appears is p_i.

Mingming's target is to collect two sets of cards, which means that for each of the n cards, Mingming should gain at least 2 copies.

Baibai wants to persuade Mingming to give up by telling him that reaching the target requires too many hours. To be scientific, Baibai asks your help to calculate the expected number of games Mingming has to win for his target. Can you help Baibai?

Input

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

For each test case:

The first line contains a single interger n $(1 \le n \le 20)$.

The second line consists of n float-point numbers $p_1, \cdots, p_n$. It's guaranteed that $\sum_i p_i = 1$ and $0 \le p_i \le 1$. Each number is accurate to three decimal places at most.

Output

For each test case, output the exptected number of games Mingming has to win. Since the answer is a rational number that can be written as $\frac{P}{Q}$, just output $P \cdot Q^{-1}$ modulo 1000000007, where Q^{-1} means the modular inverse of Q modulo 1000000007.

Sample

Input	Output
3 2 0.5 0.5 2 0.01 0.99 1 1	500000009 724329498 2

L. Teams

There are N users in a computer game. Some pairs of them are friends, while the other pairs are not. If two users are friends, and they are both online, they will invite each other, and their teams will merge into one team (Although some users in the team may not be friends). When a user logs in (becomes online), he will invite all his online friends, and their teams will be merged. When a user logs out (becomes offline), he leaves his team. When a user logs out and leaves his team, the other users in the team are still in the team, although they are not friends (But they won't become friends, just in the same team this time).

For example, there are four users A,B,C,D. A and B are friends. B and C are friends. C and D are friends. Now A, B and D are online, and C is offline. A and B are in a team, and D is in a team which contains only himself. Now if C logs in, then C will invite B and D, then the four users are in a team. Then if C logs out, A, B and D are still in a team.

Initially, all the users are offline. Now there are three kinds of operations: one logs in, one logs out, and ask how many people are there in one's team.

Input

The first line of input consists of three integers: N, M, Q. M means that there are M pairs of friends. Q means that there will be Q operations.

In the next M lines, each line contains two integers U_i and V_i, which means that U_i and V_i are friends.

In the next Q lines, each line contains a lowercase letter *op* and an integer U_i.

If *op* is 'i', then user U_i logs in. (It's guaranteed that U_i was offline before logging in)

If *op* is 'o', then user U_i logs out. (It's guaranteed that U_i was online before logging out)

If *op* is 'q', then you are asked about how many people are there in the user U_i's team. (It's guaranteed that U_i is online)

$1 \leq N, M, Q \leq 100000,\ 1 \leq U_i, V_i \leq N$

Output

For each query, output a line with an integer, denoting your answer.

Sample

Input	Output
3 2 7 1 2 2 3 i 1 i 3 q 1 i 2 q 1 o 2 q 1	1 3 2