Problem A. Expression

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Teacher Mai has n numbers $a_1, a_2, \ldots, a_n$ and $n-1$ operators (each operator is one of '+', '-' or '*') $op_1, op_2, \ldots, op_{n-1}$, which are arranged in the form $a_1\ op_1\ a_2\ op_2\ a_3 \ldots a_n$.

He wants to erase numbers one by one. In i-th round, there are $n+1-i$ numbers remained. He can erase two adjacent numbers and the operator between them, and then put a new number (derived from this one operation) in this position. After $n-1$ rounds, there is the only one number remained. The result of this sequence of operations is the last number remained.

He wants to know the sum of results of all different sequences of operations. Two sequences of operations are considered different if and only if in one round he chooses different numbers.

For example, a possible sequence of operations for $1 + 4 * 6 - 8 * 3$ is $1 + 4 * 6 - 8 * 3 \rightarrow 1 + 4 * (-2) * 3 \rightarrow 1 + (-8) * 3 \rightarrow (-7) * 3 \rightarrow -21$.

Input

First line of the input contains one integer T ($1 \le T \le 20$) — number of the test cases. Then T test cases follow.

For each test case, the first line contains one number n ($2 \le n \le 100$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$).

The third line contains a string with length $n-1$ consisting of '+', '-' and '*', which represents the operator sequence.

Output

For each test case print the answer modulo $10^9 + 7$.

Example

standard input	standard output
2	2
3	999999689
3 2 1	
-+	
5	
1 4 6 8 3	
+*-*	

Problem B. Hack It!

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Let $f(s)$ be a hash function of string s. If $s = s_0 s_1 \dots s_{n-1}$, $f(s) = (\sum\limits_{i=0}^{n-1} w(s_i) base^i) \mod r$.

Teacher Mai wants to find two different regular bracket sequences a, b with the same length $l \leq 10^4$ and the same hash value $(f(a) = f(b))$, where $w(“('') = p$, and $w(“)'') = q$.

Let us define a regular brackets sequence in the following way: Empty sequence is a regular sequence. If S is a regular sequence, then (S) is regular sequence. If A and B are regular sequences, then AB is a regular sequence.

Input

First line of the input contains integer T $(1 \leq T \leq 100)$ — number of test cases.

There are multiple test cases. All the test cases are generated randomly. For each test case, there is one line contains four numbers p, q, r, $base$ $(1 \leq p, q, r, base \leq 10^{18})$.

Output

For each test case, print two different regular bracket sequences a, b with the same length, does not exceeding 10^4 and the same hash value $f(a) = f(b)$.

Examples

standard input	standard output
`1`	`((()))`
`4 7 37 10`	`()()()`

Problem C. GCD Tree

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	64 mebibytes

Teacher Mai has a graph with n vertices numbered from 1 to n. For every $edge(u, v)$, the weight is $gcd(u, v)$. ($gcd(u, v)$ means the greatest common divisor of number u and v).

You need to find a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is maximized. Print the total weight of these edges.

Input

First line of the input contains one integer T ($1 \le T \le 10^5$) — numbef of test cases

For each test case, there is only one line containing one integer n ($1 \le n \le 10^5$).

Output

For each test case, print one integer — the answer.

Examples

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

Problem D. Too Simple

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Rhason Cheung had a simple problem, and asked Teacher Mai for help. But Teacher Mai thought this problem was too simple, sometimes naive. So she ask you for help.

Teacher Mai has m functions $f_1, f_2, \ldots, f_m : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$ (that means for all $x \in \{1, 2, \ldots, n\}$, $f(x) \in \{1, 2, \ldots, n\}$). But Rhason only knows some of these functions, and others are unknown.

She wants to know how many different function series $f_1, f_2, \ldots, f_m$ there are that for every i $(1 \leq i \leq n)$, $f_1(f_2(\ldots f_m(i))) = i$. Two function series $f_1, f_2, \ldots, f_m$ and $g_1, g_2, \ldots, g_m$ are considered different if and only if there exist i $(1 \leq i \leq m)$,$j(1 \leq j \leq n)$,$f_i(j) \neq g_i(j)$.

Input

First line of the input contains one integer T $(1 \leq T \leq 50)$.

For each test case, the first lines contains two numbers n and m $(1 \leq n, m \leq 100)$.

Then m lines follow. In i-th line, there is one number -1 or n space-separated integers. If there is only one number -1, the function f_i is unknown. Otherwise the j-th number in the i-th line means $f_i(j)$.

Output

For each test case print the answer modulo $10^9 + 7$.

Example

standard input	standard output

1
3 3
1 2 3
-1
3 2 1
	1

Problem E. Arithmetic Sequence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 seconds
Memory limit:	64 mebibytes

A sequence $b_1, b_2, \ldots, b_n$ are called (d_1, d_2)-arithmetic sequence if and only if there exist i $(1 \le i \le n)$ such that for every j $(1 \le j < i)$,$b_{j+1} = b_j + d_1$ and for every $j(i \le j < n)$, $b_{j+1} = b_j + d_2$.

Teacher Mai has a sequence $a_1, a_2, \ldots, a_n$. He wants to know how many intervals $[l, r]$ $(1 \le l \le r \le n)$ there are that $a_l, a_{l+1}, \ldots, a_r$ are (d_1, d_2)-arithmetic sequence.

Input

First line of the input contains one integer T $(1 \le T \le 15)$ — number of test cases.

For each test case, the first line contains three numbers n,d_1,d_2($1 \le n \le 10^5$,$|d_1|, |d_2| \le 1000$), the next line contains n integers $a_1, a_2, \ldots, a_n$ $(|ai| \le 10^9)$.

Output

For each test case, print the answer.

Examples

standard input	standard output

2
5 2 -2
0 2 0 -2 0
5 2 3
2 3 3 3 3
	12
	5

Problem F. Persistent Link/cut Tree

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Teacher Mai has $m+1$ trees, $T_0, T_1, \ldots, T_m$. T_0 consists of one vertex numbered 0.

He generated the T_i in next way: get a copy of T_{a_i} and T_{b_i}. Add an edge with length l_i between vertex numbered c_i in T'_{a_i} and d_i in T'_{b_i}. Relabel the vertices in the new tree. Let k be the number of vertices in T'_{a_i}. He keeps labels of vertices in T'_{a_i} the same, and adds k to labels of vertices in T'_{b_i}.

If there is a tree T with n vertices $v_0, v_1, v_2, \ldots, v_{n-1}$, $F(T) = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} d(v_i, v_j)$. ($d(v_i, v_j)$ means the distance between the v_i and v_j).

For every $i (1 \le i \le m)$, he wants to know $F(T_i)$.

Input

First line of the input contains one integer T — number of test cases ($1 \le T \le 100$).

For each test case, the first line contains one integer m ($1 \le m \le 60$), then m lines follow. The i-th line contains five numbers a_i, b_i ,c_i ,d_i, l_i ($0 \le a_i, b_i < i$, $0 \le l_i \le 10^9$). It's guarenteed that there exists a vertex numbered c_i in T_{a_i} and there exists a vertex numbered d_i in T_{b_i}.

Output

For each test case, print $F(T_i)$ modulo 10^9+7 in the i-th line.

Example

standard input	standard output
1	2
3	28
0 0 0 0 2	216
1 1 0 0 4	
2 2 1 0 3	

Problem G. Travelling Salesman Problem

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Teacher Mai is in a maze with n rows and m columns. There is a non-negative number in each cell. Teacher Mai wants to walk from the top left corner $(1, 1)$ to the bottom right corner (n, m). He can choose one direction and walk to this adjacent cell. However, he can't go out of the maze, and he can't visit a cell more than once.

Teacher Mai wants to maximize the sum of numbers in his path. And you need to print this path.

Input

First line of the input contains one integer T $(1 \leq T \leq 130)$ — number of test cases.

For each test case, the first line contains two numbers n and m $(1 \leq n, m \leq 100,\ n \cdot m \geq 2)$.

In following n lines, each line contains m numbers. The j-th number in the i-th line means the number in the cell (i, j). Every number in the cell is not more than 10^4.

Output

For each test case, in the first line, you should print the maximum sum.

In the next line you should print a string consisting of 'L','R','U' and 'D', which represents the path you find. If you are in the cell (x, y), 'L' means you walk to cell $(x, y-1)$, 'R' means you walk to cell $(x, y+1)$, 'U' means you walk to cell $(x-1, y)$, 'D' means you walk to cell $(x+1, y)$.

Examples

standard input	standard output

1
3 3
2 3 3
3 3 3
3 3 2
	25
	RRDLLDRR

Problem H. Goldbach's Conjecture

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

Let $d(x)$ be the sum of all divisors of x. x is called a good number, if every number from 1 to $d(x)$ can be expressed as a sum of distinct divisors of x.

For example, 6 is a good number, $d(6) = 1 + 2 + 3 + 6 = 12$, $4 = 1 + 3$, $5 = 2 + 3$, $7 = 1 + 6$ and so on.

Teacher Mai wants to know whether a even number p can be expressed as a sum of two good numbers.

Input

First line of the input contains one integer T $(1 \le T \le 4 \cdot 10^4)$ — number of test cases.

For each test case, there is only one line contains one even number q $(1 \le q \le 10^{18})$.

Most test cases are generated randomly.

Output

For each test case, print "`YES`" or "`NO`" in the first line. That means if is possible to express q as a sum of two good numbers.

If your answer is "`YES`", print two numbers a and b in the second lines. Both a and b should be good numbers, and $a + b = q$.

In the third and the fourth line, print the factorization of number a and b. If $a = \prod_{i=1}^{k} p_i^{e_i}$, where $p_1 < p_2 < \ldots < p_k$, p_i are all prime numbers and $e_i \ge 1$, you should print k first, then $2k$ space-seperated numbers $p_1, e_1, p_2, e_2, \ldots, p_k, e_k$.

Example

standard input	standard output

`1`
`18`
	`YES`
	`6 12`
	`2 2 1 3 1`
	`2 2 2 3 1`

Problem I. Random Inversion Machine

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 mebibytes

Teacher Mai has a game machine, which has n slots in a row, numbered from 1 to n, inclusively. He plays a game with the machine for several rounds. In each round, he divides that row into k segments with marks at boundaries between adjacent segments. Each segment must contain a positive number of slots. Then, the machine generates a random permutation of $\{1, 2, \ldots, n\}$ and displays the permutation on the slots. Finally, the machine calculates the inversion number of each segment and multiplies them together to get the score of the round. The inversion number of a sequence is the number of inversions (also called inversion pairs) in the sequence.

Teacher Mai can play the game for as many rounds as he wants, but he must use different partitions in different rounds. Two partitions are considered to be different if there is a mark somewhere in one partition but not in the other. The total game score is simply the sum of the score of each round. However, the machine has been intruded by a malware, which changes the permutation before the machine calculates the score of each round. It sorts the numbers in the m specific slots with numbers $p_1, p_2, \ldots, p_m$.

For example, if $n = 4$, $k = 1$, $m = 2$, $p = \{1, 3\}$ and the permutation generated is $(2, 4, 1, 3)$. The malware will pick numbers 2 and 1 and sort them, changing the permutation into $(1, 4, 2, 3)$. So Teacher Mai will get a score of 2 (pairs $(4, 2)$ and $(4, 3)$) in this round. Teacher Mai wants to know the maximum expected game score he can get.

Input

First line of the input contains one integer T $(0 \le T \le 10)$ — number of test cases

For each test case, the first line contains three numbers n, k and m $(1 \le k, m \le n \le 100)$.

The second line contains m numbers $p_1, p_2, \ldots, p_m$ $(1 \le p_1 < p_2 < \ldots < p_m \le n)$.

In the i-th test case, $n = 10i$.

Output

As the answer for the problem can be very large, please calculate it as an irreducible fraction A/B and output $(A \cdot B^{-1}) \mod (10^9 + 7)$ for each test case in a separate line. Here, B^{-1} is the multiplicative inverse of B modulo $10^9 + 7$. The input constraints guarantee that B and $10^9 + 7$ are coprime, so this expression is properly defined.

Examples

standard input	standard output

`1`
`10 3 4`
`2 6 7 9`
	`608333402`

Problem J. Sometimes Naive

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

Rhason Cheung had a naive problem, and asked Teacher Mai for help. But Teacher Mai thought this problem was too simple, sometimes naive. So she ask you for help.

She has a tree with n vertices, numbered from 1 to n. The weight of i-th node is w_i.

You need to support two kinds of operations: modification and query.

For a modification operation u, w, you need to change the weight of u-th node into w.

For a query operation u, v, you should output $\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} f(i,j)$. If there is a vertex on the path from u to v and the path from i to j in the tree, $f(i,j) = w_i w_j$, otherwise $f(i,j) = 0$. The number can be large, so print it modulo $10^9 + 7$.

Input

First line of the input contains one integer T $(1 \le T \le 10)$ — number of test cases.

For each test case, the first line contains two integers n and m $(1 \le n, m \le 10^5)$.

There are n integers in the next line, the i-th from them denotes w_i $(0 \le w_i \le 10^9)$.

Next $n-1$ lines contain two numbers each, u_i and v_i, that means that there is an edge between u_i and v_i.

Then m lines follow. Each line indicates an operation, and the format is "`1 u w`" for modification or "`2 u v`" for query $(0 \le w \le 10^9)$.

Output

For each test case, print the answer for each query operation.

Example

standard input	standard output
1	341
6 5	348
1 2 3 4 5 6	612
1 2	
1 3	
2 4	
2 5	
4 6	
2 3 5	
1 5 6	
2 2 3	
1 1 7	
2 2 4	