Problem A. Candies

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Rikka is poor at math. Yuta is worried about that, so he gives Rikka some math tasks to practice. One of them is described below.

There are n children and m kinds of candy. The i-th child has A_i dollars, and the unit price of the i-th kind of candy is B_i. There is an infinite supply of candy of each kind.

Each child has her favorite candy, so she will buy this kind of candy as much as possible and will not buy any candy of other kinds. For example, if a child has 10 dollars, and the unit price of her favorite candy is 4 dollars, then she will buy two candies and go home with 2 dollars left.

Yuta does not know any child's favorite candy. Now Yuta has q queries, each of them consists of an integer k. For each query, Yuta wants to know the number of pairs (i, j) $(1 \le i \le n,\ 1 \le j \le m)$ with the following property: if the i-th child's favorite candy is the j-th kind, she will take k dollars home.

To make the problem easier, it is only required to calculate the answers modulo 2. Help Rikka solve this problem for Yuta!

Input

The first line of the input contains three integers n, m and q $(1 \le n, m, q \le 5 \cdot 10^4)$.

The second line contains n integers A_i $(1 \le A_i \le 5 \cdot 10^4)$.

The third line contains m integers B_i $(1 \le B_i \le 5 \cdot 10^4)$.

The fourth line contains q integers k_i which describe the queries $(0 \le k_i < \max(B_1, B_2, \ldots, B_m))$.

It is guaranteed that $A_i \ne A_j$ and $B_i \ne B_j$ for all $i \ne j$.

Output

For each query, print a single line with a single integer: the answer to the query modulo 2.

Example

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

Problem B. Binary Strings

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

A binary string s is said to be *antisymmetric* if and only if $s[i] \neq s[|s| - i + 1]$ for all $i \in [1, |s|]$.

Yuta has n binary strings s_i, and he wants to know the number of binary antisymmetric strings of length $2L$ which contain all given strings s_i as continuous substrings. Help him find that number. As the answer can be very large, find it modulo 998 244 353.

Input

The first line of the input contains two integers n and L ($1 \leq n \leq 6$, $1 \leq L \leq 100$).

Then n lines follow, each line contains a string s_i ($1 \leq |s_i| \leq 20$) consisting of characters "0" and "1".

Output

Print a single line with a single integer: the answer modulo 998 244 353.

Examples

standard input	standard output
2 2 011 001	1
2 3 011 001	4

Note

In the second example, the strings which satisfy all the restrictions are "000111", "001011", "011001" and "100110".

Problem C. Sequence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	12 seconds
Memory limit:	512 mebibytes

Yuta has an array A_1, A_2, ..., A_n with n integers, and he keeps a copy of the initial contents of array A as A' (initially, $A'_i = A_i$). Then he executes m operations on the array A.

There are three types of operations:

- "`1` l r": Yuta wants to find the sum of A_i for all i in $[l, r]$.
- "`2` l r k": Yuta runs the following pseudocode on the sequence A:

```
for (int i = l; i <= r; i++) A[i] = A[i - k];
```

- "`3` l r": For all $i \in [l, r]$, Yuta changes A_i back to A'_i.

Help Yuta execute all the given operations.

Input

The first line of the input contains two integers n and m ($1 \leq n, m \leq 2 \cdot 10^5$).

The second line contains n integers A_i ($0 \leq A_i \leq 10^9$).

Then m lines follow, each line describes an operation in the format shown above. It is guaranteed that $1 \leq l \leq r \leq n$ and $1 \leq k < l$.

Output

For each operation of the first type, print a single line with a single integer: the required sum.

Example

standard input	standard output
5 7 1 2 3 4 5 1 1 5 2 3 4 1 1 1 5 2 3 4 2 1 1 5 3 1 5 1 1 5	15 12 11 15

Problem D. Rock-Paper-Scissors

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

Alice and Bob are going to play the famous game “Rock-Paper-Scissors”. Both of them don't like to think a lot, so both of them will use the random strategy: choose rock, paper or scissors with equal probability.

They want to play this game n times, then they will calculate the score s in the following way: if Alice won a times, Bob won b times, and the remaining $n - a - b$ games were draws, the score will be the greatest common divisor of a and b. If a or b is 0, we define the greatest common divisor of a and b as $a + b$.

Calculate the expected value of $s \cdot 3^{2n}$. Note that the answer is necessarily an integer. Because this integer may be very large, find its remainder modulo p instead.

Input

The input contains two integers n and p ($1 \le n \le 10^5$, $10^8 \le p \le 10^9$, p is prime).

Output

Print a single line with a single integer: the answer to the problem modulo p.

Examples

standard input	standard output
1 998244353	6
2 998244353	90

Problem E. Criminals

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

In an ancient country, there are $n \times m$ cities, labeled by integers from 1 to $n \cdot m$. The coordinates of the city labeled by $(x-1) \cdot m + y$ are (x, y) $(1 \le x \le n,\ 1 \le y \le m)$. There are q tourists. Initially, the i-th tourist is at city (x_i, y_i). All tourists want to go out and play in other cities.

Unfortunately, K of the $n \cdot m$ cities are controlled by criminals, so these K cities are unsafe. For safety reasons, a tourist whose initial coordinates are (x_1, y_1) can go to the city (x_2, y_2) if and only if all of the cities (x, y) $(\min(x_1, x_2) \le x \le \max(x_1, x_2),\ \min(y_1, y_2) \le y \le \max(y_1, y_2))$ are safe.

Now, for each tourist, calculate the number of cities they can reach safely (including their initial city).

Input

The first line of the input contains four integers n, m, K and q $(1 \le n, m, K, q \le 10^5)$.

Then K lines follow. Each of these lines contains two integers a_i and b_i: the coordinates of an unsafe city $(1 \le a_i \le n,\ 1 \le b_i \le m)$. It is guaranteed that each city appears at most once in this list.

Then q lines follow. Each of these lines contains two integers x_i and y_i: the initial city of each tourist $(1 \le x_i \le n,\ 1 \le y_i \le m)$. It is guaranteed that, initially, each tourist stays at a safe city.

Output

For each tourist, print a single line with a single integer: the number of cities this tourist can reach safely.

Example

standard input	standard output
4 5 4 4 1 2 2 5 3 3 4 5 1 5 2 1 2 4 4 1	3 9 8 9

Note

In the example, the third tourist can reach eight cities: $(1, 4)$, $(2, 4)$, $(3, 4)$, $(4, 4)$, $(1, 3)$, $(2, 3)$, $(2, 2)$ and $(2, 1)$.

Problem F. Build the Graph

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

For an undirected graph G with n nodes and m edges, we can define the distance $dist(i, j)$ as the length of the shortest path between nodes i and j. The length of a path is equal to the number of edges in the path. If there is no path between i and j, we set $dist(i, j)$ equal to n.

Then, we can define w_G, the weight of the graph G, as $\sum_{i=1}^{n} \sum_{j=1}^{n} \mathrm{dist}(i, j)$.

Now, given n nodes and no edges initially, we will choose no more than m pairs of nodes (i, j) $(i \neq j)$ and add an edge between the respective nodes for every chosen pair. This way, we can get an undirected graph G with n nodes and no more than m edges.

Your task is to find the minimal possible value of w_G after such construction.

Input

The first line of the input contains two integers n and m $(1 \leq n \leq 10^6,\ 1 \leq m \leq 10^{12})$.

Output

Print a single line with a single integer: the minimum possible value of w_G.

Example

standard input	standard output
4 5	14

Note

In the example, we can choose to add edges $(1, 2)$, $(1, 4)$, $(2, 4)$, $(2, 3)$ and $(3, 4)$.

Problem G. Match

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

Yuta has an undirected connected graph $G = \langle V, E\rangle$ with n nodes and $n-1$ edges. Yuta can choose some subset of edges in E and remove them. It is clear that Yuta has 2^{n-1} different subsets to remove.

Now, Yuta wants to know the number of ways to remove the edges which make the maximum matching size of the remaining graph G' divisible by m. As the answer can be very large, find its remainder modulo 998 244 353.

An edge set S is a matching of $G = \langle V, E\rangle$ if and only if each node in V is connected to at most one edge in S. The maximum matching of graph G is defined as the matching of G which has the largest size.

Input

The first line contains two integers n and m ($1 \le n \le 5 \cdot 10^4$, $1 \le m \le 200$).

Then $n-1$ lines follow, each of these lines contains two integers u and v which describe an edge in G.

Output

Print a single line with a single integer: the answer modulo 998 244 353.

Example

standard input	standard output
4 2 1 2 2 3 3 4	3

Problem H. Subsequence Sums

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Yuta has a sequence of n positive integers $A_1, \ldots, A_n$, and their sum is m. For each subsequence S of A, he calculated the sum of elements in this subsequence.

So, now Yuta has also got 2^n integers between 0 and m. For each $i \in [0, m]$, let B_i be the number of integers i he got.

Yuta shows you the array B_i, and he asks you to restore $A_1, \ldots, A_n$. If there are several possibilities, find the lexicographically smallest possible sequence.

Input

The first line of the input contains two integers n and m ($1 \leq n \leq 50$, $1 \leq m \leq 10^4$).

The second line contains $m + 1$ integers $B_0, \ldots, B_m$ ($0 \leq B_i \leq 2^n$).

Output

Print a single line with n integers $A_1, \ldots, A_n$.

It is guaranteed that there exists at least one solution. And if there are several possible solutions, print the **lexicographically smallest** one.

Examples

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

Note

In the first example, A is $[1, 2]$. A has four subsequences [], [1], [2] and [1, 2], and the sums for them are 0, 1, 2 and 3. So, $B = [1, 1, 1, 1]$.

Problem I. Nice Numbers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

In positional base-d notation, an integer $K = (A_1A_2 \dots A_m)_d$ (where $A_i \in [0, d)$ and $A_1 \neq 0$) is called *good* if and only if $A_1, \dots, A_m$ is a permutation of integers from 0 to $d - 1$.

A number K is *nice* if and only if there exists at least one $d \geq 2$ such that K is good in positional base-d notation.

Calculate the number of nice numbers in the interval $[L, R]$. As the answer may be very large, find it modulo 998 244 353.

Input

The first line of the input contains two integers L and R ($1 \leq L \leq R \leq 10^{5000}$).

Output

Print a single line with a single integer: the answer modulo 998 244 353.

Examples

standard input	standard output
5 20	3
123456 123456789	114480

Problem J. K-matching

Input file:	*standard input*
Output file:	*standard output*
Time limit:	12 seconds
Memory limit:	512 mebibytes

Consider a graph G with $n \cdot m$ nodes (i, j) $(1 \le i \le n,\ 1 \le j \le m)$. There is an edge between two nodes (a, b) and (c, d) if and only if $|a - c| + |b - d| = 1$. Each edge has a weight.

Calculate the minimum weight of a K-matching in G.

An edge set S is a matching of $G = \langle V, E \rangle$ if and only if each node in V is connected to at most one edge in S. A matching S is a K-matching if and only if $|S| = K$. The weight of a matching S is the sum of the weights of the edges in S. And finally, the minimum weight K-matching of G is defined as the K-matching of G with the minimum possible weight.

Input

The first line contains an integer t, the number of test cases ($1 \le t \le 1000$). It is guaranteed that there are at most 3 test cases with $n > 100$.

For each test case, the first line contains three integers n, m and K ($1 \le n \le 4 \cdot 10^4$, $1 \le m \le 4$, $1 \le K \le \lfloor \frac{n \cdot m}{2} \rfloor$).

Then $n - 1$ lines follow, each of these lines contains m integers $A_{i,j}$: the weights of edges between (i, j) and $(i + 1, j)$ $(1 \le A_{i,j} \le 10^9)$.

If $m > 1$, then n more lines follow, each of these lines contains $m - 1$ integers $B_{i,j}$: the weights of the edge between (i, j) and $(i, j + 1)$ $(1 \le B_{i,j} \le 10^9)$.

Output

For each test case, print a single line with a single integer: the required minimum weight.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>3
3 3 1
3 4 5
8 9 10
1 2
6 7
11 12
3 3 2
3 4 5
8 9 10
1 2
6 7
11 12
3 3 3
3 4 5
8 9 10
1 2
6 7
11 12</pre></td><td><pre>1
5
12</pre></td></tr>
</table>

Problem K. Competition

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

A wrestling competition will be held tomorrow. A total of n players will take part in it. The i-th player's strength is a_i.

If there is a match between the i-th player and the j-th player, the result will depend solely on $|a_i - a_j|$. If $|a_i - a_j| > K$, the player with the higher strength will win. Otherwise, each player will have a chance to win.

The competition rules are a little strange. For each fight, the referee will choose two players from all remaining players uniformly at random and hold a match between them. The loser will be eliminated. After $n - 1$ matches, the last remaining player will be the winner.

Given the numbers n and K and the array a, find how many players have a chance to win the competition.

Input

The first line of the input contains two integers n and K ($1 \leq n \leq 10^5$, $0 \leq K < 10^9$).

The second line contains n integers a_i ($1 \leq a_i \leq 10^9$).

Output

Print a single line with a single integer: the number of players which have a chance to win the competition.

Examples

standard input	standard output
5 3 1 5 9 6 3	5
5 2 1 5 9 6 3	1