Problem A. Atcoder Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

Print the number, modulo 998244353, of integer sequences $A = (A_1, A_2, \ldots, A_N)$ with length N that satisfy both of the following conditions.

- $0 \le A_1 \le A_2 \le \cdots \le A_N \le M$
- $A_1 \oplus A_2 \oplus \cdots \oplus A_N = X$

The problem is too easy, so output the answer for each $N = 1, 2, \ldots, NMAX$.

Input

In the first line, $NMAX, M, X$ $(1 \le NMAX \le 10^5, 0 \le M, X < 2^{60})$.

Output

$NMAX$ lines — the answers for $N = 1, 2, \ldots, NMAX$.

Examples

standard input	standard output
5 6 7	0 3 7 25 49
10 100 0	1 101 1418 38280 756912 13403840 203823022 755806367 368916768 79402702

Problem B. Best Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You are given a binary string S. You can perform the following operation any number of times:

- Replace one substring '0101' with '1010'.

What is the maximum number of operations you can perform?

Input

A binary string S $(1 \le |S| \le 5 \times 10^6)$.

Output

One integer — the answer.

Examples

standard input	standard output
`10100010011001011111`	`5`

`00000101011001101011010101100001101 00\`
`11100001001011111110010110111010100 01\`
`1110111101010101001 0101010`
`(There won't be extra line breakers \`
`in the actual test cases.)`
	`58`

Problem C. Cryptography Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

You are given m equations of the form

$$a_i \cdot x + err_i \equiv c_i \pmod p.$$

Here, err_i is an unknown random error term, chosen uniformly at random from $-\lfloor \frac{p}{200} \rfloor, \ldots, \lfloor \frac{p}{200} \rfloor$, while a_i, c_i and p are known to you.

You know that these equations hold for some unknown integer x. Find one such x.

Input

In the first line, T $(1 \le T \le 500)$ — the number of test cases. For each test case:

- In the first line, m, p $(50 \le m \le 100, 10^{15} \le p \le 10^{18})$.
- In the next m lines, a_i, c_i $(0 \le a_i, c_i \le p-1)$.
- It's guaranteed that p is a prime, a_i, x are chosen uniformly at random from 0 to $p-1$, and c_i is computed by $(a_i \cdot x + err_i) \bmod p$, err_i is an integer chosen uniformly at random from $-\lfloor \frac{p}{200} \rfloor, \ldots, \lfloor \frac{p}{200} \rfloor$.

Output

For each test case, one integer — the answer. If there are multiple solutions, you may output any.

Example

standard input	standard output
1 50 922033901407246477 492300877907148697 858503954557481 7 364781751405155O5 237143454432095134 537753813197233578 694568987600933631 ... (truncated)	578607642570710976

Note

The full sample test case is available in the contest system.

Problem D. Digit Sum Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

For a nonnegative integer x, let $f(x)$ and $g(x)$ denote the digit sum of x in binary and ternary, respectively.

Given n, a, b, c, compute

$$\left(\sum_{i=1}^{n} a^i b^{f(i)} c^{g(i)}\right) \bmod 998244353.$$

Input

In the first line, n, a, b, c $(1 \le n \le 10^{13}, 1 \le a, b, c < 998244353)$.

Output

One integer — the answer.

Examples

standard input	standard output
`123456 12345 234567 3456789`	`664963464`
`9876543210987 12816 837595 128478`	`7972694`

Problem E. Elliptic Curve Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

This problem might be well-known in some countries, but how do other countries learn about such problems if nobody poses them?

Let p be an odd prime. Compute the number of quadratic residues in $[l, r]$.

x is a quadratic residue of p iff $x^{(p-1)/2} \equiv 1 \pmod{p}$.

Input

In the first line, p, l, r $(3 \le p \le 10^{11}, 1 \le l \le r < p)$. It's guaranteed that p is an odd prime.

Output

One integer — the answer.

Examples

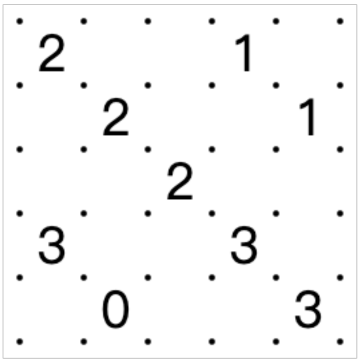
standard input	standard output
`11 3 8`	`3`
`998244353 11451400 919810000`	`454174074`
`96311898227 25437319919 55129361817`	`14846091352`

Problem F. Full Clue Problem

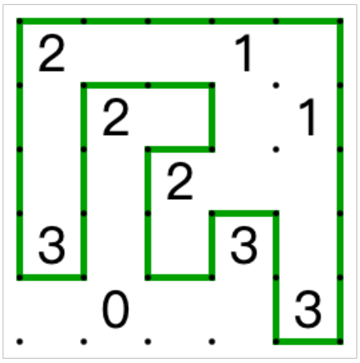
Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Slitherlink is a puzzle game played on a $n \times n$ grid. Some cells of the grid contain numbers (called *clues*). The solver must draw lines along the edges of some cells to form a loop, such that:

- The loop does not branch off or cross itself.
- The number written in a cell is equal to the number of edges surrounding the cell that are visited by the loop.



Example Problem



Solution

Construct a $n \times n$ slitherlink problem with full clues but multiple solutions. Moreover, there must be a pair of different solutions that satisfy all clues but share at most four edges.

Note: "full clues" means every cell in the problem should be filled with a clue number from $0, \ldots, 4$. "Two solutions share x edges" means that exactly x edges appear in both solutions.

Input

In the first line, n ($2 \le n \le 20$). It is guaranteed that an answer always exists.

Output

First, output an $n \times n$ matrix — the problem.

Then, output two solutions — two $n \times n$ matrices. For each cell, if it is inside the loop, output "1", otherwise output "0".

Example

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

Note

The example just shows how to output the problem and solutions. It will get a Wrong Answer verdict. These two solutions share 9 edges and the second solution doesn't satisfy all clues.

Problem G. Graph Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

You are given a directed graph with n vertices and m edges. You want to answer q queries.

For each query, you are given $k_1, p_1, p_2, \ldots, p_{k_1}$, $k_2, s_1, t_1, s_2, t_2, \ldots, s_{k_2}, t_{k_2}$. For all i $(1 \le i \le k_2)$, answer whether there is a path from s_i to t_i if $p_1, p_2, \ldots, p_{k_1}$ are deleted. Queries are independent.

Input

In the first line, n, m $(1 \le n \le 500, 0 \le m \le n(n-1))$.

In the following m lines, u, v $(1 \le u, v \le n, u \ne v)$ — a directed edge in the graph. It's guaranteed that there is no parallel edges.

In the next line, q $(1 \le q \le 4 \times 10^5)$. To make sure you answer the queries online, the input is encrypted. The input can be decrypted using the following pseudocode:

```
    cnt = 0
    for i = 1 ... q
        read(k1)
        for j = 1 ... k1
            read(p'[j])
            p[j] =(p'[j] + cnt - 1) % n + 1
        read(k2)
        for j = 1 ... k2
            read(s', t')
            s = (s' + cnt - 1) % n + 1
            t = (t' + cnt - 1) % n + 1
            cnt += query(s, t)
// if s can reach t, query return 1, otherwise, query return 0
```

In the following $2q$ lines, for each query:

- In the first line, $k_1, p'_1, \ldots, p'_{k_1}$. It's guaranteed that p_i are distinct.
- In the second line, $k_2, s'_1, t'_1, \ldots, s'_{k_2}, t'_{k_2}$. It's guaranteed that all s_i, t_i are different from all p_i.
- It's guaranteed that $1 \le k_1 \le \min(n-2, 6), \sum k_2 \le 4 \times 10^6, 1 \le p'_i, s'_i, t'_i \le n$.

Output

For each query, output a binary string with length k_2 — the answer of query(s, t) in order.

Example

standard input	standard output
5 4 1 2 2 3 3 4 4 5 2 1 4 2 1 5 1 3 3 5 3 4 1 1 2	01 1

Note

It's recommended to use Fast IO.

Problem H. Hard Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given a sequence of nonnegative integers $a_1, a_2, \ldots, a_n$. You can perform the following three types of operations any number of times.

- Choose an interval $[l, r]$, decrease all numbers in the interval by 1.
- Choose an interval $[l, r]$, decrease all numbers with odd indices in the interval by 1.
- Choose an interval $[l, r]$, decrease all numbers with even indices in the interval by 1.

Output the minimum number of operations to make all numbers equal to 0.

Input

In the first line, T $(1 \le T \le 10)$ — the number of test cases.

For each test case:

- In the first line, n $(1 \le n \le 10^5)$.
- In the second line, $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^9)$.

Output

For each test case, one integer — the answer.

Example

standard input	standard output
3 5 2 1 2 1 2 8 1000000000 1000000000 0 1000000000 \\ 1000000000 0 1000000000 1000000000 (There won't be extra line breakers \\ in the actual test cases.) 13 1 1 4 5 1 4 1 9 1 9 8 1 0	2 3000000000 19

Problem I. Interval Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given n intervals $[l_i, r_i]$. If two intervals intersect, add an undirected, unweighted edge between them.

Let $d(i, j)$ be the length of the shortest path between the i-th interval and the j-th interval. If there is no path from i to j, $d(i, j) = 0$.

For $i = 1, 2, \ldots, n$, output $\sum_{j=1}^{n} d(i, j)$.

Input

In the first line, n $(1 \le n \le 2 \times 10^5)$.

In the following n lines, l_i, r_i $(1 \le l_i < r_i \le 2n)$. It's guaranteed that the endpoints of intervals are distinct.

Output

n lines, the answer of $i = 1, 2, \ldots, n$.

Example

standard input	standard output
5 2 3 6 7 1 9 5 10 4 8	7 5 4 5 5

Problem J. Junk Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given a grid graph with n rows and m columns. Most edges are directed, which means you can walk from (x, y) to $(x+1, y)$ or $(x, y+1)$. k horizontal edges are bidirectional, which means you can walk from (x, y) to $(x, y+1)$, and $(x, y+1)$ to (x, y) too. It's guaranteed that there is no pair of bidirectional edges that share an endpoint.

You need to find l vertex-disjoint simple paths, where the i-th is from $(1, a_i)$ to (n, b_i). For a set of paths, we call a bidirectional edge *bad* if neither of its endpoints is visited by any of the paths in this set.

Output the number of all l vertex-disjoint simple paths without any bad edges, modulo 998244353.

Input

In the first line, n, m, l, k $(2 \le n, m \le 100, 1 \le l \le 50, 0 \le k \le 50)$.

In the second line, $a_1, a_2, \ldots, a_l$ $(1 \le a_1 < a_2 < \cdots < a_l \le m)$.

In the third line, $b_1, b_2, \ldots, b_l$ $(1 \le b_1 < b_2 < \cdots < b_l \le m)$.

In the following k lines, x_i, y_i $(1 \le x_i \le n, 1 \le y_i < m)$ each line, which denote that the edge (x_i, y_i) to $(x_i, y_i + 1)$ is bidirectional.

It's guaranteed that there is no pair of bidirectional edges that share an endpoint.

Output

One integer — the answer.

Examples

standard input	standard output
2 2 1 2 2 1 1 1 2 1	2
3 4 2 1 1 4 1 4 2 2	0
10 10 3 4 1 2 3 8 9 10 2 3 2 5 4 6 7 8	388035318

Problem K. Knapsack Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given $2^k - 1$ numbers $c_1, c_2, \ldots, c_{2^k-1}$ and k numbers $a_0, a_1, \ldots a_{k-1}$.

You want to find nonnegative integers $x_1, x_2, \ldots, x_{2^k-1}$ such that for all j $(0 \le j < k)$

$$\sum_{i=1}^{2^k-1} \left(\lfloor i/2^j \rfloor \bmod 2\right) x_i = a_j$$

holds and $\sum_{i=1}^{2^k-1} x_i c_i$ is maximized.

Input

In the first line, T $(1 \le T \le 100)$ — the number of test cases.

For each test case:

- In the first line, k $(2 \le k \le 4)$.
- In the second line, $c_1, c_2, \ldots, c_{2^k-1}$ $(0 \le c_i \le 10^8)$.
- In the third line, $a_0, \ldots, a_{k-1}$ $(1 \le a_i \le 10^9)$.

Output

For each test case, one integer — the answer.

Example

standard input	standard output
3 2 1 2 4 4 5 3 3226252 19270256 2430652 1187613 \\ 12496062 10286165 17494834 24 85 34 4 901133 6693218 13245489 14740234 \\ 16186210 11382783 19277321 3855635 \\ 16523184 10168517 16195031 971041 \\ 10303441 8395899 11618555 (There won't be extra line breakers \\ in the actual test cases.) 529321239 218214127 92942310 207467810	18 1949753378 7832404777567179

Problem L. Linear Congruential Generator Problem

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	1024 megabytes

Consider the following C++ code to generate a pseudorandom permutation.

```
long long x, a, b, p;

int n;
long long rand() {
        x = ((__int128)x * a + b) % p;
        return x;
}
int main() {
    cin >> n;
    cin >> x >> a >> b >> p;
    for (int i = 1; i <= n; i++) { //random shuffle [1, 2,..., n]
        perm[i] = i;
        swap(perm[i], perm[rand() % i + 1]);
    }
    for (int i = 1; i <= n; i++) { //print the result
        cout << perm[i] << (i == n ? '\n' : ' ');
    }
}
```

You are given n, a, b, p, and the output $perm_1, perm_2, \dots, perm_n$, find x $(0 \le x < p)$.

Input

In the first line, n $(n = 10^5)$.

In the second line, $perm_1, perm_2, \dots, perm_n$ $(1 \le perm_i \le n)$.

In the third line, a, b, p $(2 \le a < p, 0 \le b < p, 500 \le p \le 10^{16})$. p is a prime, a, b, x are chosen uniformly at random from their respective ranges.

Output

One integer — x. If there are multiple solutions, you may output any.

Example

standard input	standard output
100000 38898 35776 38192 80605 84959 ... 356846063 184710711 911417497 (truncated)	681779867

Note

The full sample test case is available as an attachment in the contest system.

Problem M. Minimum Element Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

We call two permutations $p_1, p_2, \ldots, p_n$ and $q_1, q_2, \ldots, q_n$ *equivalent* if and only if for every pair (i, j) $(1 \le i \le j \le n)$, the indices of the minimum element of $p_i, p_{i+1}, \ldots, p_j$ and $q_i, q_{i+1}, \ldots, q_j$ are the same.

Given x and y, consider the set of all permutations p of $\{1, 2, \ldots, n\}$ such that $p_x = y$. Find the maximum number of permutations you can pick from this set such that no two are equivalent. Output this number modulo 998244353.

The problem is too easy, so output the answer for every $y = 1, 2, \ldots, n$.

Input

In the first line n, x $(1 \le n \le 5 \times 10^5, 1 \le x \le n)$.

Output

n lines, answers for $y = 1, 2, \ldots, n$.

Examples

standard input	standard output
5 2	5 10 16 20 14
10 5	588 1176 2716 4942 7407 9101 9636 9167 7596 4862