# Problem A. Even Three is Odd

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2.5 seconds |
| Memory limit: | 512 mebibytes |

The *boboness* of a sequence of integers $(x_1, x_2, \ldots, x_n)$ is $\prod_{i=3}^{n} w(\max\{x_{i-2}, x_{i-1}, x_i\})$. Here, $1 \le x_i \le n$, and the values $w(1), w(2), \ldots, w(n)$ are given.

Bobo would like to know the sum of *boboness* of all sequences satisfying $1 \le x_i \le n$. As this sum can be very large, he is interested only in the answer modulo $(10^9 + 7)$.

## Input

The input contains zero or more test cases, and is terminated by end-of-file. For each test case:

The first line contains an integer $n$ $(3 \le n \le 2000)$.

The second line contains $n$ integers $w(1), w(2), \ldots, w(n)$ $(1 \le w(i) \le 10^9)$.

It is guaranteed that the sum of $n$ does not exceed 2000.

## Output

For each test case, output an integer which denotes the sum taken modulo $(10^9 + 7)$.

## Example

| standard input | standard output |
|---|---|
| 3<br>1 2 3<br>4<br>1 1 1 1 | 72<br>256 |

# Problem B. Walk of Length 6

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

Bobo has an undirected graph with $n$ vertices which are conveniently labeled with $1, 2, \dots, n$. Let $V$ be the set of vertices and $E$ be the set of edges. He would like to count the number of tuples $(v_1, v_2, \dots, v_6)$ where:

- $v_1, v_2, \dots, v_6 \in V$,
- $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_5, v_6\}, \{v_6, v_1\} \in E$;
- $\mathcal{C} = (\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_5, v_6\}, \{v_6, v_1\})$ is **not** a simple cycle of length 6.

## Input

The input contains zero or more test cases, and is terminated by end-of-file. For each test case:

The first line contains an integer $n$ ($1 \le n \le 1000$).

The $i$-th of the following $n$ lines contains a string $g_i$ of length $n$ where $g_{i,j}$ denotes the existence of edge $\{i, j\}$ ($g_{i,j} \in \{0, 1\}$, $g_{i,i} = 0$, $g_{i,j} = g_{j,i}$).

It is guaranteed that the sum of $n$ does not exceed 1000.

## Output

For each test case, output an integer which denotes the number of tuples.

## Example

| standard input | standard output |
|---|---|
| 3<br>011<br>101<br>110<br>4<br>0101<br>1010<br>0101<br>1010<br>6<br>011111<br>101111<br>110111<br>111011<br>111101<br>111110 | 66<br>128<br>14910 |

# Problem C. City United

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

In ICPCCamp there are $n$ cities which are conveniently labeled with $1, 2, \ldots, n$. There are also $m$ bidirectional roads: the $i$-th road connects cities $a_i$ and $b_i$.

Bobo chooses a non-empty subset of cities to form a union. For each two cities $a$ and $b$ in the union, there must exist a path from $a$ to $b$ passing through no cities outside the union. In other words, the union must be connected.

Bobo would like to know how many ways there are to choose such a subset, but he is afraid of large numbers. Therefore, he just wants to find this number modulo 2.

## Input

The first line contains two integers $n$ and $m$ ($1 \le n \le 50$, $0 \le m \le \frac{n(n-1)}{2}$).

The $i$-th of the following $m$ lines contains two integers $a_i$ and $b_i$ ($1 \le a_i, b_i \le n$, $0 < |a_i - b_i| \le 13$).

## Output

Output an integer which denotes the number of possible subsets modulo 2.

## Examples

| standard input | standard output |
|---|---|
| 3 2<br>1 2<br>2 3 | 0 |
| 3 3<br>1 2<br>2 3<br>3 1 | 1 |

# Problem D. Coins 2

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 3 seconds |
| Memory limit: | 512 mebibytes |

In ICPCCamp, people usually use coins of values $1, 2, 3, \ldots, n$.

Bobo was very poor, he had only $a_1, a_2, a_3, \ldots, a_n$ coins of values $1, 2, 3, \ldots, n$, respectively. He bought an item of an unknown value **without making change**.

The unknown item was of non-negative integer value. Find the number of possible values it may have had.

## Input

The input contains zero or more test cases, and is terminated by end-of-file. For each test case:

The first line contains one integer $n$ ($1 \le n \le 15$).

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

It is guaranteed that the number of test cases does not exceed 100, and there is at most one test case where $n > 10$.

## Output

For each test case, output an integer which denotes the number of possibilities.

## Example

| standard input | standard output |
|---|---|
| 3<br>0 1 2<br>3<br>0 2 3 | 6<br>12 |

# Problem E. Lowest Common Ancestor

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 3 seconds |
| Memory limit: | 512 mebibytes |

Bobo has a rooted tree with $n$ nodes which are conveniently labeled with $1, 2, \dots, n$. Node 1 is the root, and the $i$-th node has weight $w_i$.

He would like to find out $f(2), f(3), \dots, f(n)$ where

$$f(i) = \sum_{j=1}^{i-1} w_{\mathrm{LCA}(i,j)}.$$

## Input

The input contains zero or more test cases, and is terminated by end-of-file. For each test case:

The first line contains an integer $n$ ($2 \le n \le 2 \cdot 10^5$).

The second line contains $n$ integers $w_1, w_2, \dots, w_n$ ($1 \le w_i \le 10^4$).

The third line contains $(n-1)$ integers $p_2, p_3, \dots, p_n$, where $p_i$ denotes an edge from the $p_i$-th node to the $i$-th node ($1 \le p_i \le n$). The edges form a tree.

It is guaranteed that the sum of $n$ does not exceed $2 \cdot 10^5$.

## Output

For each test case, output $(n-1)$ integers: $f(2), f(3), \dots, f(n)$.

## Example

| standard input | standard output |
|---|---|
| 3<br>1 2 3<br>1 1<br>5<br>1 2 3 4 5<br>1 2 2 1 | 1<br>2<br>1<br>3<br>5<br>4 |

# Problem F. Multi-stage Marathon

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 3 seconds |
| Memory limit: | 512 mebibytes |

Bobo is organizing a marathon contest. The contest contains $n$ checkpoints which are conveniently labeled with $1, 2, \ldots, n$. You are given a binary matrix $G$. In this matrix, $G_{u,v} = 1$ indicates that there is a directed road from checkpoint $u$ to checkpoint $v$, and $G_{u,v} = 0$ means there is no such road.

There are $m$ players. The $i$-th player starts at checkpoint $v_i$ at moment $t_i$. As the road system is complicated, players behave quite randomly. More precisely, if at moment $t$ a player is at checkpoint $u$, at moment $(t+1)$ this player will appear at any checkpoint $v$ such that $G_{u,v} = 1$ with equal probability.

Let $E_t = P \cdot Q^{-1} \bmod (10^9 + 7)$ where $\frac{P}{Q}$ is the expected number of players at checkpoint $n$ at moment $t$, and $Q \cdot Q^{-1} \equiv 1 \mod (10^9 + 7)$. Bobo would like to know $E_1 \oplus E_2 \oplus \cdots \oplus E_T$. Note that "$\oplus$" denotes bitwise exclusive-or.

## Input

The first line contains three integers $n$, $m$ and $T$ ($1 \le n \le 70$, $1 \le m \le 10^4$, $1 \le T \le 2 \cdot 10^6$).

The $i$-th of the following $n$ lines contains a binary string $G_{i,1}, G_{i,2}, \ldots, G_{i,n}$ of length $n$. It is guaranteed that $G_{i,i} = 1$ is always true.

The $i$-th of the last $m$ lines contains two integers $t_i$ and $v_i$ ($1 \le t_1 < t_2 < \cdots < t_m \le T$, $1 \le v_i \le n$).

## Output

Output an integer which denotes the result.

## Examples

| standard input | standard output |
|---|---|
| 2 2 2<br>11<br>11<br>1 1<br>2 2 | 500000005 |
| 3 1 6<br>110<br>011<br>101<br>1 1 | 191901811 |

# Problem G. Matrix Recurrence

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 10 seconds |
| Memory limit: | 512 mebibytes |

Bobo invents a new series of matrices $M_0, M_1, \dots M_n$ defined as follows:

- $M_0 = A$,
- $M_i = \left(\prod_{j=c_i}^{i-1} M_j\right) \times B$.

Given $m \times m$ matrices $A, B$ and integers $c_1, c_2, \dots, c_n$, compute $M_n$ under $\mathbb{Z}_{\text{mod}}$ (that is, addition and multiplication of numbers are carried out modulo mod).

## Input

The input contains zero or more test cases, and is terminated by end-of-file. For each test case:

The first line contains three integers $n$, $m$ and mod ($1 \le n \le 10^6$, $1 \le m \le 5$, $2 \le \text{mod} \le 10^9$).

The $i$-th of the next $m$ lines contains $m$ integers $A_{i,1}, A_{i,2}, \dots, A_{i,m}$, and the $i$-th of the following $m$ lines contains $m$ integers $B_{i,1}, B_{i,2}, \dots, B_{i,m}$ ($0 \le A_{i,j}, B_{i,j} < \text{mod}$).

The last line contains $n$ integers $c_1, c_2, \dots, c_n$ ($0 \le c_i < i$, $c_1 \le c_2 \le \dots \le c_n$).

It is guaranteed that the sum of $n$ does not exceed $10^6$.

## Output

For each test case, output $m$ lines. On the $i$-th line, output $m$ integers $C_{i,1}, C_{i,2}, \dots, C_{i,m}$ where $C_{i,j} = M_{n,i,j}$.

## Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>2 2 1000000000
1 1
0 1
1 0
0 1
0 0
2 2 2
1 1
0 1
1 0
0 1
0 0
5 2 1000000000
1 1
0 1
1 0
0 1
0 1 2 3 4</pre></td><td><pre>1 2
0 1
1 0
0 1
1 1
0 1</pre></td></tr>
</table>

# Problem H. Permutation and noitatumreP

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

Bobo would like to count the number of permutations $(p_1, p_2, \ldots, p_n)$ of $\{1, 2, \ldots, n\}$ such that the sequence $q = (p_1, p_2, \ldots, p_n, p_n, p_{n-1}, \ldots, p_1)$ does not contain four indices $1 \le a < b < c < d \le 2n$ which satisfy $q(a) < q(c) < q(d) < q(b)$.

As this number may be very large, Bobo is only interested in its remainder modulo $(10^9 + 7)$.

## Input

The input contains zero or more test cases, and is terminated by end-of-file.

Each test case contains an integer $n$ $(1 \le n \le 10^9)$.

It is guaranteed that the number of test cases does not exceed $2 \cdot 10^4$.

## Output

For each test case, output an integer which denotes the number of ways modulo $(10^9 + 7)$.

## Example

| standard input | standard output |
|---|---|
| 4<br>1000000000 | 16<br>861159011 |

# Problem I. Compressed LCS

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

Bobo has two integer sequences $A$ and $B$, both in compressed form. $A = c_1^{a_1} c_2^{a_2} \dots c_n^{a_n}$ means that $A$ begins with $a_1$ copies of the integer $c_1$, followed by $a_2$ copies of the integer $c_2$, $a_3$ copies of the integer $c_3$, and so on. $B = d_1^{b_1} d_2^{b_2} \dots d_m^{b_m}$ is of similar format.

Bobo would like to find the LCS (longest common subsequence) for $A$ and $B$. Recall that sequence $C$ is a subsequence of $A$ if and only if $C$ can be obtained by deleting some (maybe all, maybe none) elements from $A$.

## Input

The input contains zero or more test cases, and is terminated by end-of-file. For each test case:

The first line contains two integers $n$ and $m$ ($1 \le n, m \le 2000$).

The $i$-th of the following $n$ lines contains two integers $c_i$ and $a_i$. And the $i$-th of the last $m$ lines contains two integers $d_i$ and $b_i$. The constraints are: $1 \le a_i, b_i, c_i, d_i, \sum_{i=1}^{n} a_i, \sum_{i=1}^{m} b_i \le 10^9$, $c_i \ne c_{i-1}$, $d_i \ne d_{i-1}$.

It is guaranteed that the sum of $n$ and the sum of $m$ both do not exceed 2000.

## Output

For each test case, output an integer which denotes the length of the LCS.

## Example

| standard input | standard output |
|---|---|
| 1 3<br>1 2<br>1 1<br>2 1<br>1 2<br>4 4<br>1 1<br>2 1<br>3 1<br>4 1<br>1 1<br>3 1<br>2 1<br>4 1<br>1 1<br>1000000000 999<br>1000000000 1000 | 2<br>3<br>999 |

# Problem J. Circular Sectors

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 2 seconds |
| Memory limit: | 256 mebibytes |

Bobo has drawn $n$ circular sectors on the plane. He would like to know the area of the union of all the circular sectors.

## Input

The input contains zero or more test cases, and is terminated by end-of-file. For each test case:

The first line contains an integer $n$, the number of circular sectors ($1 \le n \le 500$).

Each of the next $n$ lines contains five numbers $x_i$, $y_i$, $r_i$, $s_i$ and $\theta_i$ ($-100 \le x_i, y_i \le 100$, $1 \le r_i \le 100$, $0 \le s_i \le 6$, $0.1 \le \theta_i \le 6$). Here, $(x_i, y_i)$ is the coordinate of the circle center, $r_i$ is the radius of the circle, $s_i$ is the starting angle in radians (counter-clockwise from the positive direction of the $x$ axis) and $\theta_i$ is the central angle in radians (this means that the sector arc goes from angle $s_i$ to angle $s_i + \theta_i$ where the angle is measured counter-clockwise from the positive direction of the $x$ axis). Also, $x_i$, $y_i$ and $r_i$ are integers, and $s_i$ and $\theta_i$ are real numbers with exactly 3 digits after the decimal point.

It is guaranteed that the sum of $n$ does not exceed 500.

## Output

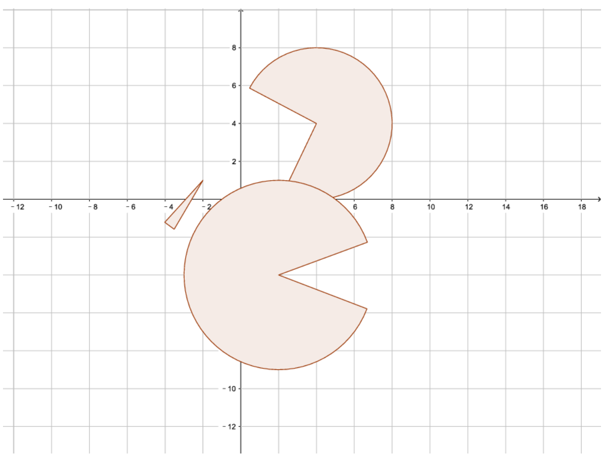
For each test case, output a real number denoting the answer. Your answer will be considered correct if its relative or absolute error doesn't exceed $10^{-6}$.

## Example

| standard input | standard output |
|---|---|
| 2<br>-3 -5 5 0.705 0.217<br>-5 1 4 3.070 4.136<br>1<br>-4 -4 1 0.485 2.260<br>3<br>4 4 4 4.266 4.673<br>2 -4 5 0.353 5.565<br>-2 1 3 3.974 0.207 | 35.80050000000000700000<br>1.12999999999999940000<br>106.44493143870359000000 |

## Note

The image below shows the third test case.



# Problem K. Welcome to ICPCCamp 2017

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

ICPCCamp teams are often selected by a mysterious $(X, Y)$-rule described in a blog (?).

There are $(n+1)$ selection contests held to choose *ICPCCamp team* among $m$ teams conveniently labeled with $1, 2, \ldots, m$. The number of teams attending the $i$-th contest is $k_i$. As the last (the $(n+1)$-th) contest called EasyCamp-Final is very important, $k_{n+1} = m$ always holds. The scoreboard of the $i$-th contest is $r_{i,1}, r_{i,2}, \ldots, r_{i,k_i}$ which indicates that team $r_{i,j}$ has rank $j$ in the contest.

The $(X, Y)$-rule works as follows. Firstly, two non-negative integers $X$ and $Y$ and a permutation $P = \{p_1, p_2, \ldots, p_n\}$ of $\{1, 2, \ldots, n\}$ are chosen. After that, the first $X + Y$ distinct teams in the list $\{r_{n+1,1}, r_{n+1,2}, \ldots, r_{n+1,Y}, r_{p_1,1}, r_{p_2,1}, \ldots, r_{p_n,1}, r_{p_1,2}, r_{p_2,2}, \ldots, r_{p_n,2}, \ldots\}$ will be selected as *ICPCCamp team*. In other words, the list goes in the following order: the first $Y$ EasyCamp-Final teams, then the top teams from the first $n$ contests in the order defined by $P$, then the second teams from the first $n$ contests in the same order, and so on.

Bobo would like to know the number of possible sets of *ICPCCamp team*s modulo $(10^9 + 7)$ if he can choose $X$, $Y$ and $P$ arbitrarily.

Wish you enjoy yourself in the upcoming World Finals!

## Input

The input contains zero or more test cases, and is terminated by end-of-file. For each test case:

The first line contains two integers $n$ and $m$ ($0 \le n \le 2 \cdot 10^5$, $1 \le m \le 2 \cdot 10^5$).

The $i$-th of following $n$ lines contains an integer $k_i$ followed by $k_i$ integers $r_{i,1}, r_{i,2}, \ldots, r_{i,k_i}$ ($1 \le k_i \le m$).

The last line contains $m$ integers $r_{n+1,1}, r_{n+1,2}, \ldots, r_{n+1,m}$ ($1 \le r_{i,j} \le m$, and for each $i$, the numbers $\{r_{i,1}, r_{i,2}, \ldots, r_{i,k_i}\}$ are distinct).

It is guaranteed that both the sum of $k_i$ and the sum of $m$ do not exceed $2 \cdot 10^5$.

## Output

For each test case, output an integer which denotes the number of sets modulo $(10^9 + 7)$.

## Example

| standard input | standard output |
|---|---|
| 2 3<br>2 1 3<br>3 2 1 3<br>2 1 3<br>0 3<br>1 2 3 | 5<br>4 |