The 3rd Universal Cup

Stage 11: Sumiyosi

October 5-6, 2024

This problem set should contain 15 problems (A to O) on 24 numbered pages.

Problem A. Welcome to NPCAPC

Time limit: 4 seconds
Memory limit: 1024 megabytes

Among strings of length N consisting of uppercase and lowercase English letters, find the number of strings that contain both 'NPCAPC' and 'npcapc' as subsequences (not necessarily contiguous), modulo 998244353.

You have T test cases to solve.

Constraints

- $1 \leq T \leq 5000$
- $1 \leq N \leq 10^9$

Input

The input is given in the following format from standard input:

T
case_1
case_2
$\vdots$
case_T

Here, case_i denotes the i-th test case. Each test case is given in the following format:

N

Output

Output T lines. On the i-th line, output the answer for the i-th test case.

Examples

standard input	standard output
4 12 6 5839 123456	924 0 966252995 432934749
3 123456789 987654321 999999999	333574957 124462731 163251704

Note

For the first sample case:

In the first test case, there are 924 strings that satisfy the conditions, such as 'npcapcNPCAPC' and 'NPCnpcAapPCc'.

Problem B. Some Sum of Subset

Time limit: 2 seconds
Memory limit: 1024 megabytes

You are given a sequence of positive integers $A = (A_1, A_2, \ldots, A_N)$ of length N. For $k = 0, 1, \ldots, N$, solve the following problem.

> Find the number of subsets S of $\{1, 2, \ldots, N\}$ that satisfy the following condition, modulo 998244353.
>
> - There exists a subset T of S such that $|T| = |S| - k$ and $\sum_{i \in T} A_i \geq M$.

Constraints

- $1 \leq N \leq 3000$
- $1 \leq M \leq 3000$
- $1 \leq A_i \leq 3000$

Input

The input is given in the following format from standard input:

N M
A_1 A_2 $\ldots$ A_N

Output

Output $N + 1$ lines. In the i-th line ($1 \leq i \leq N + 1$), output the answer for $k = i - 1$.

Examples

standard input	standard output
4 7 3 1 5 2	6 4 1 0 0
1 5 7	1 0
9 18 1 9 5 6 2 7 1 4 8	346 309 230 126 46 10 1 0 0 0

Note

For the first sample case:

As an example, let's explain the case when $k = 1$.

- For $S = \{1, 3, 4\}$, if we let $T = \{3, 4\}$, then $|T| = |S| - 1$ and $\sum_{i \in T} A_i \geq 7$, so it satisfies the condition.

Other subsets satisfying the condition are $S = \{1, 2, 3\}, \{2, 3, 4\}, \{1, 2, 3, 4\}$, totaling 3 subsets. Therefore, when $k = 1$, the answer is 4.

Problem C. Solve with Friends

Time limit: 2 seconds
Memory limit: 1024 megabytes

Namuka and Napuka have decided to solve all N problems, namely problem 1, problem 2, ..., problem N.

Initially, their tiredness are both 0, but solving a problem increases the tiredness of the person who solved it by 1. When solving Problem i with a current tiredness of j, Namuka-kun takes $A_i + C_j$ minutes, and Napuka-kun takes $B_i + D_j$ minutes. The two cannot solve problems simultaneously.

Find the minimum total time required for Namuka and Napuka to solve all N problems.

Constraints

- $1 \le N \le 2 \times 10^5$
- $1 \le A_i, B_i, C_i, D_i \le 10^9$

Input

The input is given in the following format from standard input:

N

$A_1\ A_2\ \ldots\ A_N$

$B_1\ B_2\ \ldots\ B_N$

$C_0\ C_1\ \ldots\ C_{N-1}$

$D_0\ D_1\ \ldots\ D_{N-1}$

Output

Output the answer.

Examples

standard input	standard output
3 1 3 5 6 4 2 1 2 3 1 2 3	10
5 2 4 3 1 2 9 2 5 3 8 1 2 8 3 2 5 4 3 2 1	28
8 21 85 72 22 81 20 88 28 75 22 78 92 55 56 73 44 39 14 64 27 73 42 16 84 27 7 91 85 69 95 70 27	621

Note

For the first sample case:

When Namuka solves problem 1 and problem 2 in order, and Napuka solves problem 3, the total time taken can be calculated as follows:

- Namuka solves problem 1. Namuka's current tiredness is 0, so it takes $A_1 + C_0 = 1 + 1 = 2$ minutes. Namuka's tiredness increases by 1.
- Namuka solves problem 2. Namuka's current tiredness is 1, so it takes $A_2 + C_1 = 3 + 2 = 5$ minutes. Namuka's tiredness increases by 1.
- Napuka solves problem 3. Napuka's current tiredness is 0, so it takes $B_2 + D_0 = 2 + 1 = 3$ minutes. Napuka's tiredness increases by 1.

Therefore, the total time is $2 + 5 + 3 = 10$ minutes, which is the minimum.

Problem D. Two Box

Time limit: 6 seconds
Memory limit: 1024 megabytes

You are given a sequence of non-negative integers $A = (A_1, A_2, \ldots, A_N)$ of length N and Q queries. The i-th query is described as follows:

- Change A_{x_i} to y_i, and then compute the answer to the following problem based on the updated sequence A.

There are two boxes, one white and one black, and M balls numbered from 1 to M. Initially, all balls are in the white box.
You perform the following operation N times:

- Choose an integer x that satisfies $1 \le x \le M$. Move ball x from its current box to the other box.

After the i-th operation, all the numbers on the balls in the black box must be less than or equal to A_i. Compute the number of possible sequences of operations that satisfy this condition, modulo 998244353.

Process the queries in order.

Constraints

- $1 \le N, Q \le \mathbf{3 \times 10^4}$
- $1 \le M \le 15$
- $1 \le x_i \le N$
- $1 \le A_i, y_i \le M$

Input

The input is given in the following format from standard input:

N M Q
A_1 A_2 $\ldots$ A_N
x_1 y_1
x_2 y_2
$\vdots$
x_Q y_Q

Output

Output Q lines. On the i-th line, output the answer to the i-th query.

Examples

standard input	standard output
3 3 2 1 3 1 3 2 1 3	5 14
6 8 1 3 8 7 7 1 6 1 4	2100
12 10 8 1 3 2 6 3 6 7 7 5 5 4 7 12 4 7 10 4 2 9 8 9 9 8 3 4 9 10 2	2741280 3007680 1503840 1916160 1972800 728640 1821600 621440

Note

For the first sample case:

For the first query, $A = (1, 3, 2)$. In this case, possible sequences of operations include, for example:

- Choose $x = 1$. Move ball 1 from the white box to the black box. The black box now contains ball 1.
- Choose $x = 3$. Move ball 3 from the white box to the black box. The black box now contains balls 1 and 3.
- Choose $x = 3$. Move ball 1 from the black box back to the white box. The black box now contains ball 1.

Other possible sequences of x are $(1, 1, 1)$, $(1, 1, 2)$, $(1, 2, 1)$, and $(1, 2, 2)$, totaling 4 additional possibilities. Therefore, there are 5 possible sequences of operations.

Problem E. Aim High

Time limit: 2 seconds
Memory limit: 1024 megabytes

You will play a game on a 2-dimensional plane. Initially, at each lattice point (x, y) where $-100 \le x \le 100$ and $-100 \le y \le 0$, one piece is placed.

You can perform the following operation zero or more times:

- Choose two points (a, b) and (c, d) where $|a - c| + |b - d| = 1$. Move one piece from (a, b) by rotating it 90 degrees clockwise or counterclockwise around (c, d), and remove one piece from (c, d).

Your goal is to perform operations such that, after all operations, there is at least one piece at a point with a y-coordinate of at least N. Determine whether it is possible to achieve the goal, and if so, construct a sequence of operations.

You are given T test cases. Solve each test case accordingly.

Constraints

- $1 \le T \le 6$
- $1 \le N \le 6$

Input

The input is given in the following format from standard input:

T
case_1
case_2
$\vdots$
case_T

Here, case_i denotes the i-th test case. Each test case is given in the following format:

N

Output

For each of the T test cases, output the results in the given order, separated by newlines.

For each test case, if it is impossible to achieve the goal, output '-1'. Otherwise, first output the number of operations K, followed by K lines describing the operations. For the i-th operation, when moving a piece from (a_i, b_i) by rotating it 90 degrees around (c_i, d_i) to (e_i, f_i), output as follows:

K
$a_1\ b_1\ c_1\ d_1\ e_1\ f_1$
$a_2\ b_2\ c_2\ d_2\ e_2\ f_2$
$\vdots$
$a_K\ b_K\ c_K\ d_K\ e_K\ f_K$

Example

standard input	standard output
1 1	1 1 0 0 0 0 1

Note

In the first operation, a piece at $(1, 0)$ is rotated 90 degrees clockwise around $(0, 0)$ and placed at $(0, 1)$.

This operation allows placing a piece at the point $(0, 1)$, where the y-coordinate is at least 1, thus achieving the goal.

Problem F. Train Seats

Time limit: 3 seconds
Memory limit: 1024 megabytes

There are N people numbered from 1 to N sitting on M chairs arranged in a row. The chair in the i-th position from the left is called chair i. Person i sits on chair A_i.

When a person sits down, let L and R be the numbers of the closest occupied chairs to the left and right of that person, respectively (if there is no such chair on the left, $L = 0$; if there is no such chair on the right, $R = M + 1$). The score of the person is calculated as $R - L$.

There are $N!$ possible ways for the N people to sit in order. Find the maximum possible total sum of the scores of all N people.

Constraints

- $1 \le N \le 2 \times 10^5$
- $N \le M \le 10^9$
- $1 \le A_i \le M$
- If $i \ne j$, then $A_i \ne A_j$

Input

The input is given in the following format from standard input:

N M

A_1 A_2 $\ldots$ A_N

Output

Output the answer.

Examples

standard input	standard output
3 10 3 7 10	28
5 20 3 10 11 14 17	73
10 1000000000 136909656 243332691 <...> 182482400 (download in the attchments)	7649951260

Note

For the first sample case:

For example, if the people sit in the order of person 3, person 1, and then person 2, the scores are as follows:

- When person 3 sits down, $L = 0$ and $R = 11$, so their score is $11 - 0 = 11$.
- When person 1 sits down, $L = 0$ and $R = 10$, so their score is $10 - 0 = 10$.
- When person 2 sits down, $L = 3$ and $R = 10$, so their score is $10 - 3 = 7$.

Therefore, the total sum of scores is $11 + 10 + 7 = 28$, which is the maximum.

Problem G. Many Common Segment Problems

Time limit: 8 seconds
Memory limit: 1024 megabytes

PCT has created the following problem.

Common Segment
You are given N segments $[L_1, R_1], [L_2, R_2], \ldots, [L_N, R_N]$. Here, $[L, R]$ represents the set of all integers from L to R inclusive.
There are $2^N - 1$ ways to choose one or more segments, among these, find the number of ways where the intersection of all chosen segments is non-empty. Output the result modulo 998244353.

PCT accidentally lost some of the L_i and R_i values in the test cases. To help him out, solve the following problem.

Many Common Segment Testcases
You are given test cases for **Common Segment**. However, the missing L_i, R_i values are replaced with '-1'.
It is known that the original test cases satisfied $1 \le L_i \le R_i \le M$ $(1 \le i \le N)$. For all possible original test cases, solve **Common Segment** and find the sum of all answers modulo 998244353.

Constraints

- $1 \le N, M \le 10^5$
- $L_i = -1$ or $1 \le L_i \le M$
- $R_i = -1$ or $1 \le R_i \le M$
- If $L_i, R_i \ge 1$, then $L_i \le R_i$

Input

The input is given in the following format from standard input:

N M
L_1 R_1
L_2 R_2
$\vdots$
L_N R_N

Output

Output the answer.

Examples

standard input	standard output
3 3 1 -1 2 2 2 3	18
5 8 1 7 2 3 4 8 6 8 1 5	15
10 13 4 -1 -1 -1 7 11 -1 -1 -1 -1 -1 -1 11 -1 3 8 -1 9 -1 -1	841024210

Note

For the first sample case:

All possible test cases and their corresponding answers for **Common Segment** are as follows:

- When $(L_i, R_i) = (1, 1), (2, 2), (2, 3)$, the answer is 4.
- When $(L_i, R_i) = (1, 2), (2, 2), (2, 3)$, the answer is 7.
- When $(L_i, R_i) = (1, 3), (2, 2), (2, 3)$, the answer is 7.

Therefore, the total answer is $4 + 7 + 7 = 18$.

Problem H. Music Game

Time limit: 2 seconds
Memory limit: 1024 megabytes

There are N switches numbered from 1 to N. Currently, all switches are off. You will press the switches one by one in any order you choose, but each switch is broken. Specifically, pressing switch i takes T_i seconds and behaves as follows:

- With probability $\frac{A_i}{B_i}$, it turns on.
- With probability $1 - \frac{A_i}{B_i}$, all N switches turn off.

Whether a switch turns on or not is independently determined each time it is pressed. Additionally, you cannot press another switch while pressing one.

Your goal is to turn all switches on as quickly as possible. When you press the switches appropriately, find the expected number of seconds required to turn all switches on, modulo 998244353.

Constraints

- $1 \le N \le 2 \times 10^5$
- $1 \le T_i \le 10^6$
- $1 \le A_i \le B_i \le 10^6$

Input

The input is given in the following format from standard input:

N

T_1 A_1 B_1

T_2 A_2 B_2

$\vdots$

T_N A_N B_N

Output

It can be proven that the expected value is always a rational number. Moreover, under the constraints of this problem, it can also be proven that when this value is expressed as a reduced fraction $\frac{P}{Q}$, $Q \neq 0$ (mod 998244353). Therefore, there exists a unique integer R satisfying $R \times Q = P \pmod{998244353}$ and $0 \le R < 998244353$. Output this R.

Examples

standard input	standard output
2 3 3 5 2 4 7	831870305
5 2 5 9 6 4 7 1 9 14 17 8 13 10 4 11	914017655
8 6 2 8 3 1 8 5 30 71 7 9 58 6 4 7 6 9 25 2 8 67 6 6 55	923892723

Note

For the first sample case:

As an example of a sequence of operations, the following exists (this sequence does not necessarily represent the optimal operations):

- Press switch 1 over 3 seconds. Switch 1 turns on.
- Press switch 2 over 2 seconds. All switches turn off.
- Press switch 2 over 2 seconds. Switch 2 turns on.
- Press switch 1 over 3 seconds. Switch 1 turns on.

In this sequence, the time taken is 10 seconds, and the probability that the operations proceed in this way is $\frac{3}{5} \times \frac{3}{7} \times \frac{4}{7} \times \frac{3}{5} = \frac{108}{1225}$.

Additionally, in this case, the expected number of seconds required to turn all switches on when pressing switches appropriately is $\frac{65}{6}$ seconds.

Problem I. Left Equals Right

Time limit: 2 seconds
Memory limit: 1024 megabytes

Find the number of permutations $(P_1, \ldots, P_N)$ of $(1, \ldots, N)$ that satisfy the following condition, modulo 998244353.

- There exists an integer i $(1 \le i < N)$ such that $A_{P_1} + \cdots + A_{P_i} = A_{P_{i+1}} + \cdots + A_{P_N}$.

Constraints

- $2 \le N \le 100$
- $1 \le A_i \le 100$

Input

The input is given in the following format from standard input:

N

A_1 A_2 $\ldots$ A_N

Output

Output the answer.

Examples

standard input	standard output
3 4 9 5	4
2 100 100	2
8 3 2 6 3 1 2 4 5	11520

Note

For the first sample case:

There are $3!(= 6)$ permutations of $(1, 2, 3)$, of which 4 satisfy the condition:

- $(1, 3, 2)$
- $(2, 1, 3)$
- $(2, 3, 1)$
- $(3, 1, 2)$

For example, for $(1, 3, 2)$, choosing $i = 2$, we have $A_1 + A_3 = A_2 = 9$, which satisfies the condition.

Problem J. Again Permutation Problem

Time limit: 5 seconds
Memory limit: 1024 megabytes

You are given M permutations of $(1, 2, \ldots, N)$. The i-th permutation is $P_i = (P_{i,1}, P_{i,2}, \ldots, P_{i,N})$.

You have a sequence $Q = (1, 2, \ldots, N)$. You can perform the following operation zero or more times:

- Choose an integer i satisfying $1 \leq i \leq M$, and update Q to $(Q_{P_{i,1}}, Q_{P_{i,2}}, \ldots, Q_{P_{i,N}})$.

Find the sum of the inversion number for all possible sequences Q that can be obtained after performing any number of operations. Output the result modulo 998244353.

Constraints

- $1 \leq N \leq 30$
- $1 \leq M \leq 30$
- $P_i = (P_{i,1}, P_{i,2}, \ldots, P_{i,N})$ is a permutation of $(1, 2, \ldots, N)$.

Input

The input is given from standard input in the following format:

N M
$P_{1,1}$ $P_{1,2}$ $\ldots$ $P_{1,N}$
$P_{2,1}$ $P_{2,2}$ $\ldots$ $P_{2,N}$
$\vdots$
$P_{M,1}$ $P_{M,2}$ $\ldots$ $P_{M,N}$

Output

Output the answer.

Examples

standard input	standard output
3 2 1 2 3 2 3 1	4
5 2 3 4 5 1 2 1 5 4 3 2	50
30 12 1 2 9 4 5 6 <...> 26 3 28 29 30 (download in the attchments)	701414999

Note

For the first sample case:

There are three possible sequences Q: $(1, 2, 3)$, $(2, 3, 1)$, and $(3, 1, 2)$. Their inversion numbers are 0, 2, and 2, respectively, so the answer is $0 + 2 + 2 = 4$.

Problem K. Peace with Magic

Time limit: 2 seconds
Memory limit: 1024 megabytes

The NPCA country consists of N squares arranged in a straight line, numbered from 1 to N from left to right. Let the height of square i be H_i. Initially, $H_1 = H_2 = \cdots = H_N = 0$.

For each $1 \le i \le N-1$, if the absolute difference between H_i and H_{i+1} is less than D_i, a conflict arises between square i and square $i+1$. Napuka-kun, the peace-loving king of NPCA country, aims to eliminate all conflicts between every pair of adjacent squares. To achieve this, Napuka-kun can perform the following magic any number of times (including zero):

- Choose integers i and j such that $1 \le i \le j \le N$ and $H_i = H_{i+1} = \cdots = H_j$, then add 1 to each of $H_i, H_{i+1}, \ldots, H_j$.

Determine the minimum number of magic Napuka-kun needs to perform to achieve his goal.

Constraints

- $2 \le N \le 100$
- $0 \le D_i \le 1000$

Input

The input is given in the following format from standard input:

N

$D_1\ D_2\ \ldots\ D_{N-1}$

Output

Output the answer.

Examples

standard input	standard output
4 2 3 1	4
3 0 0	0
10 1 9 5 6 2 7 1 4 8	22

Note

For the first sample case:

Initially, $(H_1, H_2, H_3, H_4) = (0, 0, 0, 0)$. For example, the magic can be cast as follows:

- Choose $(i, j) = (1, 3)$. Then $(H_1, H_2, H_3, H_4) = (1, 1, 1, 0)$.
- Choose $(i, j) = (1, 2)$. Then $(H_1, H_2, H_3, H_4) = (2, 2, 1, 0)$.
- Choose $(i, j) = (2, 2)$. Then $(H_1, H_2, H_3, H_4) = (2, 3, 1, 0)$.
- Choose $(i, j) = (2, 2)$. Then $(H_1, H_2, H_3, H_4) = (2, 4, 1, 0)$.

Napuka-kun casts the magic 4 times to achieve the goal, and this is the minimum number of casts. Note that you may choose $i = j$.

Problem L. Construction of Town

Time limit: 2 seconds
Memory limit: 1024 megabytes

You are given a non-decreasing sequence of positive integers $X = (X_1, X_2, \ldots, X_{N-1})$ of length $N-1$.

Define the cost of a simple connected undirected graph G with N vertices and M edges as $\sum_{i=1}^{N} \sum_{j=i+1}^{N} X_{d(i,j)}$. Here, $d(i,j)$ is defined as the minimum number of edges one must traverse to move from vertex i to vertex j in G.

Construct one simple connected undirected graph G with N vertices and M edges that minimizes the cost.

Constraints

- $2 \le N \le 100$
- $N-1 \le M \le \frac{N(N-1)}{2}$
- $1 \le X_1 \le X_2 \le \cdots \le X_{N-1} \le 10^9$

Input

The input is given in the following format from standard input:

N M

X_1 X_2 $\ldots$ X_{N-1}

Output

When the i-th edge in the graph connects vertex A_i and vertex B_i, output M lines as follows:

A_1 B_1

A_2 B_2

$\vdots$

A_M B_M

Examples

standard input	standard output
3 2 4 5	1 2 1 3
4 6 12 34 56	1 2 1 3 1 4 2 3 2 4 3 4

Note

For the first sample case:

In this output, the cost is $X_{d(1,2)} + X_{d(1,3)} + X_{d(2,3)} = X_1 + X_1 + X_2 = 13$.

Since there is no undirected graph with 3 vertices and 2 edges whose cost is 12 or less, this output is correct.

Problem M. Admired Person

Time limit: 2 seconds
Memory limit: 1024 megabytes

Namuka has a sequence of integers $A = (A_1, A_2, \ldots, A_N)$ of length N, and Namuka's ideal person has a sequence $B = (B_1, B_2, \ldots, B_M)$ of length M.

To get closer to their ideal person, Namuka selects M distinct elements from A, arranges them in any order, and forms a sequence $C = (C_1, C_2, \ldots, C_M)$ of length M.

Find the minimum possible value of $\sum_{i=1}^{M} |B_i - C_i|$.

Constraints

- $1 \le M \le N \le 5000$
- $1 \le A_i, B_i \le 10^9$

Input

The input is given in the following format from standard input:

N M

A_1 A_2 $\ldots$ A_N

B_1 B_2 $\ldots$ B_M

Output

Output the answer.

Examples

standard input	standard output
5 3 2 6 5 1 1 6 3 8	4
3 2 1 1 9 1 1	0
11 7 13 21 9 5 16 32 15 29 20 40 4 24 34 43 39 18 30 11	32

Note

For the first sample case:

For example, by choosing $C = (6, 2, 5)$, the minimum value $|6 - 6| + |3 - 2| + |8 - 5| = 4$ can be achieved.

Problem N. Product Matrix

Time limit: 3 seconds
Memory limit: 1024 megabytes

The time limit for this problem might be tight.

You are given an $N \times N$ square matrix $P(x)$, where each element is a first-degree polynomial. The (i,j)-th element of $P(x)$ is $a_{i,j}x + b_{i,j}$.

Compute each coefficient $c_0, c_1, \ldots, c_M$ of the $(1,1)$-element $f(x) = \sum_{i=0}^{M} c_i x^i$ of the product $\prod_{i=0}^{M-1} P(2^i x) = P(x)P(2x)\ldots P(2^{M-1}x)$, modulo (10^9+7).

Constraints

- $1 \le N \le 6$
- $1 \le M \le 5 \times 10^5$
- $0 \le a_{i,j}, b_{i,j} < 10^9+7$

Input

The input is given in the following format from standard input:

N M

$a_{1,1}$ $a_{1,2}$ $\ldots$ $a_{1,N}$

$a_{2,1}$ $a_{2,2}$ $\ldots$ $a_{2,N}$

$\vdots$

$a_{N,1}$ $a_{N,2}$ $\ldots$ $a_{N,N}$

$b_{1,1}$ $b_{1,2}$ $\ldots$ $b_{1,N}$

$b_{2,1}$ $b_{2,2}$ $\ldots$ $b_{2,N}$

$\vdots$

$b_{N,1}$ $b_{N,2}$ $\ldots$ $b_{N,N}$

Output

Output the coefficients $c_0, c_1, \ldots, c_M$ modulo (10^9+7), each on a separate line in this order.

Examples

standard input	standard output
2 2 1 2 3 4 2 0 1 2	4 8 14
(download in the attchments)	(download in the attchments)
(download in the attchments)	(download in the attchments)

Note

For the first sample case:

Since

$$P(x)P(2x) = \begin{pmatrix} x+2 & 2x \\ 3x+1 & 4x+2 \end{pmatrix} \begin{pmatrix} 2x+2 & 4x \\ 6x+1 & 8x+2 \end{pmatrix} = \begin{pmatrix} 14x^2+8x+4 & 20x^2+12x \\ 30x^2+24x+4 & 44x^2+28x+4 \end{pmatrix},$$

the answer is $f(x) = 14x^2 + 8x + 4$.

Problem O. New School Term

Time limit: 3 seconds
Memory limit: 1024 megabytes

There are $2N$ students at NPCA School, and each student is assigned a unique number from 1 to $2N$. Napuka-kun is a teacher at NPCA School and needs to divide the students into **two classes of N students each**.

The dissatisfaction of the class division is defined as follows:

- For each integer i ($1 \leq i \leq M$), if student A_i and student B_i are in the same class, add 2^i to the total dissatisfaction.

Construct one way of class division that minimizes the dissatisfaction for Napuka-kun.

Constraints

- $1 \leq N \leq 5000$
- $0 \leq M \leq 10^6$
- $1 \leq A_i < B_i \leq 2N$
- If $i \neq j$, then $(A_i, B_i) \neq (A_j, B_j)$
- All input values are integers

Input

The input is given from standard input in the following format:

N M
A_1 B_1
A_2 B_2
$\vdots$
A_M B_M

Output

Output should be in the following format:

$S_1 S_2 \ldots S_{2N}$

Here, S_i is either '0' or '1', indicating which class student i belongs to.

If there are multiple valid class divisions, you may output any one of them.

Examples

standard input	standard output
2 4 1 3 2 4 1 4 1 2	0101
3 7 2 5 1 3 4 6 2 6 4 5 2 4 5 6	001101

Note

For the first sample case:

When dividing into a class consisting of students 1 and 3, and another class consisting of students 2 and 4, the dissatisfaction is calculated as follows:

- For $i = 1$, students 1 and 3 are in the same class.
- For $i = 2$, students 2 and 4 are in the same class.
- For $i = 3$, students 1 and 4 are in different classes.
- For $i = 4$, students 1 and 2 are in different classes.

Thus, the total dissatisfaction for this division is $2^1 + 2^2 = 6$, which is the minimum. You may output '1010'.

If you divide as '0111', the dissatisfaction is 4, but the classes do not have exactly N students each, so it does not satisfy the conditions.