Problem A. XOR Tree Path

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

You are given a rooted tree with N vertices, labeled from 1 to N, where vertex 1 is the root. The i-th edge $(1 \le i \le N-1)$ connects vertices U_i and V_i.

Each vertex of the tree is painted either white or black. Vertex i $(1 \le i \le N)$ is painted white if $A_i = 0$, and black if $A_i = 1$.

You can perform the following operation any number of times (possibly zero):

- Choose a **leaf** vertex x, and flip the color (change white vertices to black and black vertices to white) of all vertices on the path from the root to the vertex x (including the root and the vertex x) .

Your goal is to maximize the number of black vertices. What is the maximum number of black vertices that can be achieved?

Input

The input is given from Standard Input in the following format:

$$\begin{array}{l} N \\ A_1\ A_2\ \cdots\ A_N \\ U_1\ V_1 \\ U_2\ V_2 \\ \vdots \\ U_{N-1}\ V_{N-1} \end{array}$$

- All values in the input are integers.
- $2 \le N \le 10^5$
- $0 \le A_i \le 1$ $(1 \le i \le N)$
- $1 \le U_i, V_i \le N$ $(1 \le i \le N-1)$
- The given graph is a tree.

Output

Output the maximum number of black vertices that can be achieved by performing any number of operations.

Examples

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

Note

In the first example, it is possible to make all vertices black by performing the following operations.

1. choose vertex 2 and perform the operation. This makes vertex 1 white and vertex 2 black.
2. choose vertex 5 and perform the operation. This makes vertex $1, 3, 5$ black.

Problem B. Magical Wallet

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

You have a magical wallet with X yen in it. (Yen is the currency of Japan.)

Using the magic on this wallet, you can rearrange the amount of money in the wallet as a decimal string in any order you like. For example, if you have a magical wallet with 120 yen, you can use magic to change the amount of money in the wallet to any of the following: 12 yen, 21 yen, 102 yen, 120 yen, 201 yen, or 210 yen (leading zeros are ignored).

You will now visit N shops with the magical wallet **in order**. At the i-th shop ($1 \le i \le N$), a product costing A_i yen is sold, and if the magical wallet contains at least A_i yen, you can pay A_i yen from the magical wallet to buy the product.

You can use magic as much as you like whenever you want. How many products can you buy at most?

Input

The input is given from Standard Input in the following format:

N X
A_1 A_2 $\cdots$ A_N

- All values in the input are integers.
- $1 \le N \le 100$
- $1 \le X < 10^4$
- $1 \le A_i < 10^4$ $(1 \le i \le N)$

Output

Print the answer.

Examples

standard input	standard output
2 120 142 90	2
1 119 911	1
5 1000 900 90 900 9 900	3
7 1171 6328 2419 8302 7503 1744 8495 1522	5

Note

In the first sample, you can buy two products by doing the following:

1. Use magic to change the amount of money in the wallet from 120 yen to 201 yen.
2. Buy a product for 142 yen at the first shop. The amount of money in the wallet becomes $201 - 142 = 59$ yen.

3. Use magic to change the amount of money in the wallet from 59 yen to 95 yen.
4. Buy a product for 90 yen at the second shop. The amount of money in the wallet becomes $95-90=5$ yen.

Problem C. Parallel Processing (Easy)

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

This is the easy version of the problem. The only difference between two versions is the constraint of N.

You are given a mysterious monoid $(M, \oplus)$ and 4 CPUs to compute it.

Compute the cumulative $\oplus$ of a sequence $A = (A_1, A_2, \ldots, A_N)$ of M in parallel using 4 CPUs, minimizing the number of operations.

Statement

You are given an integer N. Write a program in a custom language to do the following and minimize the number of instructions in your program.

Specification

This program can handle 2004 variables $A[1], A[2], \ldots, A[2000], C_1, C_2, C_3, C_4$. Each variable can hold a sequence of integers, and $A[i]$ $(1 \leq i \leq 2000)$ is initialized to $A[i] = (i)$. (Here (i) denotes an integer sequence consisting of one i.)

At the end of the execution, the following condition must be satisfied:

- For each of $i = 1, 2, ..., N$, $A[i] = (1, 2, ..., i)$ holds.

Format

The first line of the program contains an integer L representing the number of instructions in the program.

The L instructions are written in 4 lines per instruction from the 2nd to the $(4L + 1)$-th lines, and are executed sequentially from top to bottom.

Each instruction is written as 12 integers $c_1, a_1, b_1, c_2, a_2, b_2, c_3, a_3, b_3, c_4, a_4, b_4$, where each integer must be between 1 and 2000 (inclusive).

For each instruction, the following operations are performed in order:

1. Assigns concat$(A[a_1], A[b_1])$ to C_1.
2. Assigns concat$(A[a_2], A[b_2])$ to C_2.
3. Assigns concat$(A[a_3], A[b_3])$ to C_3.
4. Assigns concat$(A[a_4], A[b_4])$ to C_4.
5. Assigns C_1 to $A[c_1]$.
6. Assigns C_2 to $A[c_2]$.
7. Assigns C_3 to $A[c_3]$.
8. Assigns C_4 to $A[c_4]$.

Here, concat(x, y) denotes the sequence obtained by concatenating the sequences x and y in that order.

Input

The input is given in the following format:

N

- All values in the input are integers.
- $2 \le N \le 16$

Output

Let L be the minimum number of instructions. Output in the following format:

L op_1 op_2 $\vdots$ op_L

op_i $(1 \le i \le L)$ represents the i-th operation and should be output in the following format:

$c_1\ a_1\ b_1$ $c_2\ a_2\ b_2$ $c_3\ a_3\ b_3$ $c_4\ a_4\ b_4$

Here, each integer must be between 1 and 2000.

Examples

standard input	standard output
2	1 2 1 2 2000 2000 2000 2000 2000 2000 2000 2000 2000
4	2 2 1 2 4 3 4 2000 2000 2000 2000 2000 2000 3 2 3 4 2 4 2000 2000 2000 2000 2000 2000

Note

In the first example, the first operation changes $A[2]$ to $(1, 2)$ and $A[2000]$ to $(2000, 2000)$. At the end of the execution, $A[1] = (1)$ and $A[2] = (1, 2)$, which satisfies the specification.

Problem D. Parallel Processing (Hard)

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

This is the hard version of the problem. The only difference between two versions is the constraint of N.

You are given a mysterious monoid $(M, \oplus)$ and 4 CPUs to compute it.

Compute the cumulative $\oplus$ of a sequence $A = (A_1, A_2, \ldots, A_N)$ of M in parallel using 4 CPUs, minimizing the number of operations.

Statement

You are given an integer N. Write a program in a custom language to do the following and minimize the number of instructions in your program.

Specification

This program can handle 2004 variables $A[1], A[2], \ldots, A[2000], C_1, C_2, C_3, C_4$. Each variable can hold a sequence of integers, and $A[i]$ $(1 \leq i \leq 2000)$ is initialized to $A[i] = (i)$. (Here (i) denotes an integer sequence consisting of one i.)

At the end of the execution, the following condition must be satisfied:

- For each of $i = 1, 2, ..., N$, $A[i] = (1, 2, ..., i)$ holds.

Format

The first line of the program contains an integer L representing the number of instructions in the program.

The L instructions are written in 4 lines per instruction from the 2nd to the $(4L + 1)$-th lines, and are executed sequentially from top to bottom.

Each instruction is written as 12 integers $c_1, a_1, b_1, c_2, a_2, b_2, c_3, a_3, b_3, c_4, a_4, b_4$, where each integer must be between 1 and 2000 (inclusive).

For each instruction, the following operations are performed in order:

1. Assigns $\text{concat}(A[a_1], A[b_1])$ to C_1.
2. Assigns $\text{concat}(A[a_2], A[b_2])$ to C_2.
3. Assigns $\text{concat}(A[a_3], A[b_3])$ to C_3.
4. Assigns $\text{concat}(A[a_4], A[b_4])$ to C_4.
5. Assigns C_1 to $A[c_1]$.
6. Assigns C_2 to $A[c_2]$.
7. Assigns C_3 to $A[c_3]$.
8. Assigns C_4 to $A[c_4]$.

Here, $\text{concat}(x, y)$ denotes the sequence obtained by concatenating the sequences x and y in that order.

Input

The input is given in the following format:

N

- All values in the input are integers.
- $17 \le N \le 1000$

Output

Let L be the minimum number of instructions. Output in the following format:

L
op_1
op_2
$\vdots$
op_L

op_i $(1 \le i \le L)$ represents the i-th operation and should be output in the following format:

c_1 a_1 b_1
c_2 a_2 b_2
c_3 a_3 b_3
c_4 a_4 b_4

Here, each integer must be between 1 and 2000.

Problem E. Five Med Sum

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

You are given five integer sequences $A = (A_1, \ldots, A_N)$, $B = (B_1, \ldots, B_N)$, $C = (C_1, \ldots, C_N)$, $D = (D_1, \ldots, D_N)$, and $E = (E_1, \ldots, E_N)$ of length N.

Find the following value modulo 998244353.

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N}\sum_{m=1}^{N} \mathrm{med}(A_i, B_j, C_k, D_l, E_m)$$

Here, $\mathrm{med}(a, b, c, d, e)$ represents the median of a, b, c, d, e.

Input

The input is given from Standard Input in the following format:

$$\begin{array}{l} N \\ A_1\ A_2\ \cdots\ A_N \\ B_1\ B_2\ \cdots\ B_N \\ C_1\ C_2\ \cdots\ C_N \\ D_1\ D_2\ \cdots\ D_N \\ E_1\ E_2\ \cdots\ E_N \end{array}$$

- All values in the input are integers.
- $1 \le N \le 10^5$
- $0 \le A_i, B_i, C_i, D_i, E_i < 998244353$ $(1 \le i \le N)$

Output

Print the answer.

Examples

standard input	standard output
1 1 2 3 4 5	3
3 1 2 3 1 3 2 2 1 3 2 3 1 3 1 2	486

Problem F. Forestry

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

You are given a tree T with N vertices and $N-1$ edges. The vertices are numbered from 1 to N, and the edges are numbered from 1 to $N-1$. Edge i $(1 \leq i \leq N-1)$ connects vertices U_i and V_i. Additionally, an integer A_v is written on vertex v $(1 \leq v \leq N)$.

There are 2^{N-1} ways to choose some of the edges of the tree. For each choice, **the score** is defined as follows:

- Let G be the graph obtained by removing the unselected edges from T. For each connected component of G, consider the minimum value of the integers written on its vertices, and sum those minimum values. The sum is the score.

Find the sum of the scores over all choices modulo 998244353.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
A_1\ A_2\ \cdots\ A_N \\
U_1\ V_1 \\
U_2\ V_2 \\
\vdots \\
U_{N-1}\ V_{N-1}
\end{array}
$$

- All values in the input are integers.
- $2 \leq N \leq 3 \times 10^5$
- $1 \leq A_i \leq 10^9$ $(1 \leq i \leq N)$
- $1 \leq U_i, V_i \leq N$ $(1 \leq i \leq N-1)$
- The given graph is a tree.

Output

Output the answer.

Examples

standard input	standard output
4 1 2 3 4 1 2 2 4 3 2	44
5 3 5 6 5 1 4 1 2 3 3 5 1 3	154

Problem G. Range NEQ

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

You are given two positive integers N and M.

Count the number of permutations $P = (P_0, P_1, \ldots, P_{NM-1})$ of $(0, 1, \ldots, NM-1)$ such that the following condition is satisfied, modulo 998244353.

- For all integers i such that $0 \le i < NM$, $\left\lfloor \frac{i}{M} \right\rfloor \ne \left\lfloor \frac{P_i}{M} \right\rfloor$ holds.

Input

The input is given from Standard Input in the following format:

N M

- All values in the input are integers.
- $2 \le N \le 1000$
- $1 \le M \le 1000$

Output

Output the answer.

Examples

standard input	standard output
2 2	4
5 1	44
167 91	284830080

Problem H. Expanded Hull

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

You are given integers N, K and N points $(X_1, Y_1, Z_1), \dots, (X_N, Y_N, Z_N)$ in three-dimensional space.

Let V be the convex hull of the N points $(KX_1, KY_1, KZ_1), \dots, (KX_N, KY_N, KZ_N)$. Count the number of points that are contained in the interior or on the boundary of V and have integer coordinates, modulo 998244353.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N\ K \\
X_1\ Y_1\ Z_1 \\
\vdots \\
X_N\ Y_N\ Z_N
\end{array}
$$

- All values in the input are integers.
- $4 \le N \le 100$
- $1 \le K \le 10^{15}$
- $-200 \le X_i, Y_i, Z_i \le 200$ $(1 \le i \le N)$
- $(X_i, Y_i, Z_i) \ne (X_j, Y_j, Z_j)$ $(1 \le i < j \le N)$
- There is no plane passing through all N points.

Output

Output the answer.

Examples

standard input	standard output
4 2 0 0 0 1 0 0 0 1 0 0 0 1	10
4 10000 0 0 0 1 0 0 0 1 0 0 0 1	59878050
8 314159265358979 5 -3 -3 -5 -3 -3 0 5 -3 0 0 10 4 2 6 -4 2 6 0 -5 6 0 0 -5	152811018

Note

In the first example, there are 10 points that are contained in the interior or on the boundary of V and have integer coordinates: $(0,0,0)$, $(1,0,0)$, $(2,0,0)$, $(0,1,0)$, $(1,1,0)$, $(0,2,0)$, $(0,0,1)$, $(1,0,1)$, $(0,1,1)$, $(0,0,2)$

Problem I. Peaceful Results

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

Alice, Bob, and Chris will play rock-paper-scissors N times. However, each of them has the following restrictions on what hand they can play:

- Alice must play rock exactly A_R times, paper exactly A_P times, and scissors exactly A_S times.
- Bob must play rock exactly B_R times, paper exactly B_P times, and scissors exactly B_S times.
- Chris must play rock exactly C_R times, paper exactly C_P times, and scissors exactly C_S times.

Alice, Bob, and Chris are very good friends, so they want to make sure that they tie every single game over the N rounds. Count the number of ways to choose the hands of the three players for the N rounds that achieves this, modulo 998244353.

Input

The input is given from Standard Input in the following format:

N
A_R A_P A_S
B_R B_P B_S
C_R C_P C_S

- All values in the input are integers.
- $1 \leq N \leq 1.5 \times 10^6$
- $0 \leq A_R, A_P, A_S, B_R, B_P, B_S, C_R, C_P, C_S \leq 1.5 \times 10^6$
- $A_R + A_P + A_S = B_R + B_P + B_S = C_R + C_P + C_S = N$

Output

Output the answer.

Examples

standard input	standard output
2 2 0 0 1 1 0 1 0 1	2
3 0 1 2 3 0 0 1 1 1	0
333333 111111 111111 111111 111111 111111 111111 111111 111111 111111	383902959

Problem J. Make Convex Sequence

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

You are given two integer sequences $L = (L_1, L_2, \ldots, L_N)$ and $R = (R_1, R_2, \ldots, R_N)$. Determine if there exists a sequence $A = (A_1, A_2, \ldots, A_N)$ of **real numbers** that satisfy the following conditions:

- For all integers i such that $1 \le i \le N$, $L_i \le A_i \le R_i$ holds.
- For all integers i such that $2 \le i \le N-1$, $A_{i-1} + A_{i+1} \ge 2A_i$ holds.

Input

The input is given from Standard Input in the following format:

N
$L_1\ L_2\ \cdots\ L_N$
$R_1\ R_2\ \cdots\ R_N$

- All values in the input are integers.
- $3 \le N \le 3 \times 10^5$
- $1 \le L_i \le R_i \le 10^9$ $(1 \le i \le N)$

Output

If there exists a sequence A of real numbers that satisfies the conditions, output `Yes`. Otherwise, output `No`.

Examples

standard input	standard output
4 2 1 2 5 4 6 5 8	Yes
3 1 4 2 3 7 4	No

Note

In the first example, for example, $A = (4, \frac{3}{2}, 3, 7)$ satisfies the conditions.

Problem K. Count Arithmetic Progression

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

You are given two sequences of integers $L = (L_1, L_2, \ldots, L_N)$ and $R = (R_1, R_2, \ldots, R_N)$, find the number of sequences $A = (A_1, A_2, \ldots, A_N)$ of integers that satisfy the following conditions, modulo 998244353:

- For all integers i such that $1 \le i \le N$, $L_i \le A_i \le R_i$ holds.
- Let $d = A_2 - A_1$. For all integers i such that $1 \le i \le N - 1$, $A_{i+1} - A_i = d$ holds.

Input

The input is given from Standard Input in the following format:

N
$L_1\ L_2\ \cdots\ L_N$
$R_1\ R_2\ \cdots\ R_N$

- All values in the input are integers.
- $2 \le N \le 3 \times 10^5$
- $1 \le L_i \le R_i \le 10^{12}$ $(1 \le i \le N)$

Output

Print the number of sequences A that satisfy the conditions, modulo 998244353.

Examples

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

Note

In the first example, there are 6 sequences that satisfy the conditions: $(5, 5, 5)$, $(5, 6, 7)$, $(6, 5, 4)$, $(6, 6, 6)$, $(7, 5, 3)$, $(7, 6, 5)$.

Problem L. Many Products

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

You are given two positive integers N and M and a sequence of integers $A = (A_1, A_2, \ldots, A_N)$. Let $\boldsymbol{X}$ be the set of all N-tuples of positive integers $(x_1, x_2, \ldots, x_N)$ that satisfy $\left(\prod_{i=1}^{N} x_i\right) = M$.

Find the following value modulo 998244353.

$$\sum_{(x_1,x_2,\ldots,x_N)\in \boldsymbol{X}} \prod_{i=1}^{N}(x_i + A_i)$$

Input

The input is given from Standard Input in the following format:

N M
A_1 A_2 $\cdots$ A_N

- All values in the input are integers.
- $1 \le N \le 2 \times 10^5$
- $1 \le M \le 10^{12}$
- $0 \le A_i < 998244353$ $(1 \le i \le N)$

Output

Output the answer.

Examples

standard input	standard output
2 3 0 1	10
5 1 0 1 2 3 4	120
10 314159265358 0 1 2 3 4 5 6 7 8 9	658270849

Note

In the first sample, $\boldsymbol{X} = \{(1, 3), (3, 1)\}$, and the answer is $(1 + 0)(3 + 1) + (3 + 0)(1 + 1) = 10$.

Problem M. Colorful Graph

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	64 megabytes

You are given a directed graph with N vertices and M edges. The vertices are numbered from 1 to N, and the edges are numbered from 1 to M. Edge i $(1 \le i \le M)$ goes from vertex A_i to vertex B_i.

Your task is to color each vertex of the graph with one of the colors $1, \ldots, N$ in such a way that the following conditions are satisfied:

- For each vertex i $(1 \le i \le N)$, let c_i be the color assigned to it. For any pair (i, j) $(1 \le i < j \le N)$ such that $c_i = c_j$, there exists a path from vertex i to vertex j or from vertex j to vertex i (or both).
- The value of $\max\{c_1, \ldots, c_N\}$ is as small as possible.

Construct one coloring that satisfies these conditions.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N\ M \\
A_1\ B_1 \\
A_2\ B_2 \\
\vdots \\
A_M\ B_M
\end{array}
$$

- All values in the input are integers.
- $1 \le N \le 7 \times 10^3$
- $0 \le M \le 7 \times 10^3$
- $1 \le A_i, B_i \le N$ $(1 \le i \le M)$
- $A_i \ne B_i$ $(1 \le i \le M)$
- $(A_i, B_i) \ne (A_j, B_j)$ $(1 \le i < j \le M)$

Output

Output the color assignment $c_1, c_2, \ldots, c_N$ that satisfies these conditions.

Examples

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

Note

The memory limit for this problem is 64 MB.

Problem N. XOR Reachable

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

You are given integers N, M, and K, and an undirected graph with N vertices and M edges. The vertices are numbered from 1 to N, and the edges are numbered from 1 to M. Edge i ($1 \le i \le M$) connects vertices A_i and B_i, and has a non-negative integer weight C_i. The graph may have multiple edges, but there are no self-loops.

You are given Q queries. In the i-th ($1 \le i \le Q$) query, you are given an integer D_i. Find the number of pairs of integers (u, v) that satisfy the following conditions:

- $1 \le u < v \le N$
- It is possible to move from vertex u to vertex v by only using edges j such that $(C_j \oplus D_i) < K$, where $\oplus$ denotes the bitwise XOR operation.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N\ M\ K \\
A_1\ B_1\ C_1 \\
A_2\ B_2\ C_2 \\
\vdots \\
A_M\ B_M\ C_M \\
Q \\
D_1 \\
D_2 \\
\vdots \\
D_Q
\end{array}
$$

- All values in the input are integers.
- $2 \le N \le 10^5$
- $1 \le M \le 10^5$
- $0 \le K < 2^{30}$
- $1 \le A_i < B_i \le N$ ($1 \le i \le M$)
- $0 \le C_i < 2^{30}$ ($1 \le i \le M$)
- $1 \le Q \le 10^5$
- $0 \le D_i < 2^{30}$ ($1 \le i \le Q$)

Output

Output Q lines. The i-th line should contain the answer to the i-th query.

Examples

standard input	standard output
4 5 5 1 2 17 1 3 4 2 3 20 2 4 3 3 4 5 4 0 7 16 167	2 6 3 0
9 13 488888932 2 7 771479959 3 8 783850182 5 7 430673756 6 8 350738034 4 9 400768807 2 3 83653266 1 2 829786563 5 8 357613791 7 9 579696618 3 7 423191200 3 5 867380255 1 9 907715012 6 9 1033650694 8 498260055 144262908 117665696 848664012 983408133 32610599 478007408 134182829	16 7 5 13 13 16 16 5

Problem O. Jewel Game

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

You are given a directed graph with N vertices and M edges. The vertices are numbered from 1 to N, and the edges are numbered from 1 to M. Edge i ($1 \leq i \leq M$) goes from vertex U_i to vertex V_i. The graph may have self-loops, but there are no multiple edges. It is guaranteed that there is at least one edge going out from each vertex.

There are K jewels placed on some of the vertices. The i-th ($1 \leq i \leq K$) jewel is located at vertex X_i and has value W_i.

Alice and Bob are playing a game using this graph. At the beginning of the game, Alice is at vertex A, and Bob is at vertex B. Starting with Alice, Alice and Bob take turns performing the following action:

- Choose one edge going out from the vertex they are currently on and move to the next vertex along that edge. If the next vertex has a jewel, they take the jewel and remove it from the graph.

The game ends when either all jewels are taken, or the current state of the game (turn, positions of both players, remaining jewels) has appeared before. At the end of the game, the score of the game is defined as:

$$\text{(the sum of the values of the gems taken by Alice)} - \text{(the sum of the values of the gems taken by Bob)}$$

Alice wants to maximize this score, while Bob wants to minimize it. Find the score of the game when both players play optimally.

Input

The input is given from Standard Input in the following format:

N M A B
U_1 V_1
$\vdots$
U_M V_M
K
X_1 W_1
$\vdots$
X_K W_K

- All values in the input are integers.
- $2 \leq N \leq 30$
- $1 \leq A, B \leq N$
- $1 \leq U_i, V_i \leq N$ $(1 \leq i \leq M)$
- $(U_i, V_i) \neq (U_j, V_j)$ $(1 \leq i < j \leq M)$
- For every x $(1 \leq x \leq N)$, there exists at least one i such that $U_i = x$.
- $1 \leq K \leq 10$
- $1 \leq X_1 < \cdots < X_K \leq N$

- $X_i \notin \{A, B\}$ $(1 \le i \le K)$
- $1 \le W_i \le 10^8$ $(1 \le i \le K)$

Output

Print the score of the game when both players play optimally.

Examples

standard input	standard output
5 16 1 1 1 2 1 3 1 4 1 5 2 3 2 4 2 5 3 2 3 4 3 5 4 2 4 3 4 5 5 2 5 3 5 4 4 2 4 3 84 4 38 5 96	46
8 16 8 4 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 1 1 5 2 6 3 7 4 8 5 1 6 2 7 3 8 4 6 1 29 2 34 3 41 5 7 6 26 7 94	-23