acm International Collegiate Programming Contest 2013 IBM event sponsor

Maratona de Programação da SBC 2013

Sub-Regional Brasil do ACM ICPC

14 de Setembro de 2013

Caderno de Problemas

Informações Gerais

Este caderno contém 10 problemas; as páginas estão numeradas de 1 a 19, não contando esta página de rosto. Verifique se o caderno está completo.

A) Sobre a entrada

1) A entrada de seu programa deve ser lida da *entrada padrão*.
2) A entrada é composta de um único caso de teste, descrito em um número de linhas que depende do problema.
3) Quando uma linha da entrada contém vários valores, estes são separados por um único espaço em branco; a entrada não contém nenhum outro espaço em branco.
4) Cada linha, incluindo a última, contém o caractere final-de-linha.
5) O final da entrada coincide com o final do arquivo.

B) Sobre a saída

1) A saída de seu programa deve ser escrita na *saída padrão*.
2) Quando uma linha da saída contém vários valores, estes devem ser separados por um único espaço em branco; a saída não deve conter nenhum outro espaço em branco.
3) Cada linha, incluindo a última, deve conter o caractere final-de-linha.

Promoção:

SBC Sociedade Brasileira de Computação

Patrocínio:

Fundação Carlos Chagas

v1.0

Problem A

Zero or One

File: `zero.[c|cpp|java]`

Everyone probably knows the game *Zero or One* (in some regions in Brazil also known as *Two or One*), used to determine a winner among three or more players. For those unfamiliar, the game works as follows. Each player chooses a value between zero or one; prompted by a command (usually one of the contestants announces "Zero or... One!"), all participants show the value chosen using a hand: if the value chosen is one, the contestant shows a hand with an extended index finger; if the value chosen is zero, the contestant shows a hand with all fingers closed. The winner is the one who has chosen a value different from all others. If there is no player with a value different from all others (e.g. all players choose zero, or some players choose zero and some players choose one), there is no winner.

Alice, Bob and Clara are great friends and play Zerinho all the time: to determine who will buy popcorn during the movie session, who will enter the swimming pool first, etc.. They play so much that they decided make a plugin to play Zerinho on Facebook. But since the don't know how to program computers, they divided the tasks among friends who do know, including you.

Given the three values chosen by Alice, Bob and Clara, each value zero or one, write a program that determines if there is a winner, and in that case determines who is the winner.

Input

The input contains a single line, with three integers A, B and C ,indicating respectively the values chosen by Alice, Beto and Clara.

Output

Your program must output a single line, containing a single character. If Alice is the winner the character must be '`A`', if Beto is the winner the character must be '`B`', if Clara is the winner the character must be '`C`', and if there is no winner the character must be '`*`' (asterisc).

Restrictions

- $A, B, C \in \{0, 1\}$

Examples

Input	**Output**
`1 1 0`	`C`

Input	**Output**
`0 0 0`	`*`

Input	**Output**
`1 0 0`	`A`

Problem B

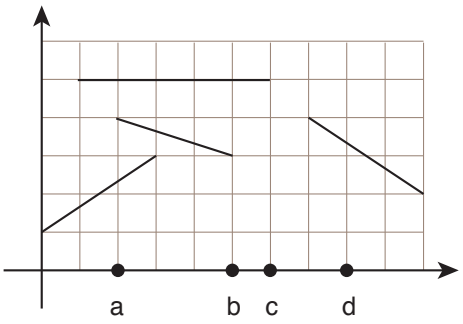
Balloon

File: `balloon.[c|cpp|java]`

One of the main difficulties of organizing a Programming Contest is collecting the balloons that escape and are trapped on the roof of the competition hall: often the contract with the hall owner requires that the hall must be cleaned soon after the event, otherwise a fine is applied.

This year the organization of our competition have been more prudent: it got the project design of the ceiling of the hall, and wants your help to determine what will happen to a loose balloon depending on the position on the ground where it is released (that is, whether it is blocked by the ceiling or escapes to the outside of the hall).

The ceiling of the hall consists of several plans that, viewed from the side, can be described by line segments, as shown in the figure below:



The balloon may be considered to be a point. When a balloon touches a line segment that is horizontal, it gets stuck. When a balloon touches a segment that is tilted, the balloon glides to the highest point of the segment and escapes. It may then escape from the hall or it may touch more segments. There are no points in common between the line segments that form the ceiling.

For example, if a balloon is released at the positions marked as `a` or `b`, it will be stuck in the position with coordinates $(2, 5)$; if a balloon is released at the position marked `c`, it will be stuck in the position of coordinates $(6, 5)$; and if the balloon is released at the position marked as `d`, it will not be blocked and will escape out of the hall in the position of coordinate $x = 7$.

Write a program that, given the description of the ceiling of the hall as line segments, answers a series of queries about the final positions of balloons released at the hall floor.

Input

The first line of input contains two integers N and C indicating, respectively, the number of segments describing the ceiling, and the number of queries. Each of the next N lines contains four integers X_1, Y_1, X_2, Y_2, describing a line segment from the ceiling, with end points at coordinates (X_1, Y_1) and (X_2, Y_2).

Each of the next C describe a query and contains an integer X, indicating that the query wants to determine what happens to a balloon releaset at the point of coordinates $(X, 0)$.

Output

For each query in the input your program must output a single line. If the balloon escapes the hall, the line must contain a single integer X, indicating the x coordinate where eht balloon escapes the hall. Otherwise, the line must contain two integers X and Y indicating the position (x, y) where the balloon gets stuck in the ceiling.

Restrictions

- $1 \leq N \leq 10^5$
- $1 \leq C \leq 10^5$
- $0 \leq X_1, X_2 \leq 10^6$, $0 < Y_1, Y_2 \leq 10^6$, $X_1 \neq X_2$
- no two x coordinate values are equal, considering all segments.
- $0 \leq X \leq 10^6$

Examples

Input	**Output**
4 4 0 1 3 3 1 5 6 5 5 3 2 4 7 4 10 2 2 5 8 6	2 5 2 5 7 6 5

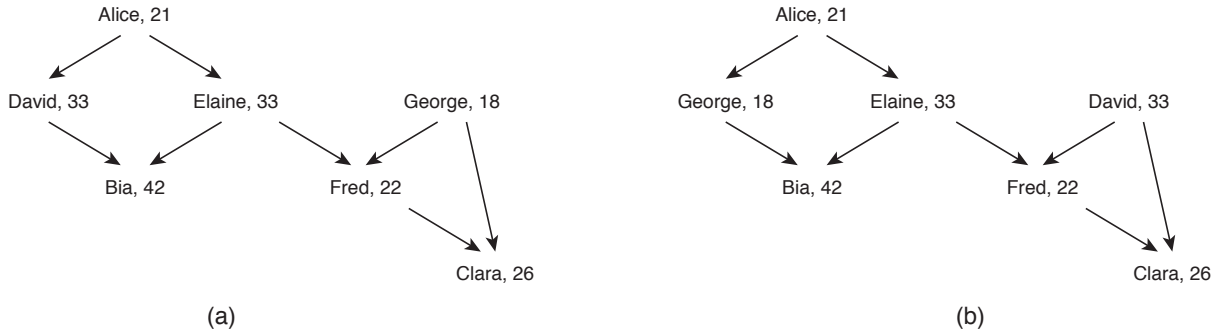
Input	**Output**
4 3 1 3 4 2 10 3 7 4 2 3 8 3 3 5 5 4 4 9 8	1 7 8 3

Problem C
Boss

File: `boss.[c|cpp|java]`

Everyone knows Iks, the last trend in social network, which made so much success that competitors like Facebook and Google+ are strugling to keep their own social networks in business. As several ".com" companies, Iks started in a small garage, but today employs hundreds of thousands.

Iks has also a non-standard management system. For example, there are no committees or boards, which means less time spent on internal meetings. However, as usual in other companies, there is a chain (or rather several chains) of command, as a person may manage other employees, and may be managed by other employees. The figure below shows the chain of command for some employees, along with their ages.



A person P_1 may manage another person P_2 directly (when P_1 is the immediate superior of P_1) or indirectly (when P_1 manages direclty a person P_3 who manages P_2 directly or indirectly). For example, in the figure above, Alice manages David directly and Claire indirectly. A person does not manage himself/herself, either directly or indirectly.

One folklore that developed in Wall Street is that Iks is so successfull because in its chain of command a manager is always younger than the managed employee. As we can see in figure above, that is not true. But this folklore prompted Iks to develop a tool to study its own management system, and to understand whether age plays a role in its success. You have been hired to work on that tool.

Given the description of the chain of command at Iks and the ages of its employees, write a program that answers a series of instructions. Instructions are of two types: management change and query. An instruction of management change swaps the positions of two employees A and B. As an example, figure (b) above shows the resulting chain of command when David and George change their respective positions in the chain of command. A query instruction names one employee A and asks the age of the youngest manager of A; that is, the youngest person in the chain of command at Iks that manages A, either directly or indirectly. For example, in figure (a) above the youngest manager of Clara is 18 years old; in figure (b), the youngest manager of Clara is 21 years old.

Input

Each test case is described using several lines. The first line of a test case contains three integers N, M and I, representing respectively the number of employees, the number of direct manage relations and the number of instructions. Employees are identified by numbers from 1 to N. The second line contains N integers K_i, where K_i indicates the age of the employee number i.

Each of the following M lines contains two integers X and Y, indicating that X manages Y directly. Then it follows I lines, each describing an instruction. An instruction of type management change is

described by a line containing the identifier `T` followed by two integers A and B, indicating the two employers that must swap places in the chain of command. An instruction of type query is described by a line containing the identifier `P` followed by one integer E, representing the number of an employer. The last instruction is of type query.

Output

For each instruction of type query your program must print a single line containing a single integer, the age of the employee who is the youngest person that manages the employer named in the query (directly or indirectly), if that person exists. If the employee does not have a manager, print an `*` (asterisc).

Restrictions

- $1 \leq N \leq 500$
- $0 \leq M \leq 60 \times 10^3$
- $1 \leq I \leq 500$
- $1 \leq K_i \leq 100$, for $1 \leq i \leq N$
- $1 \leq X, Y \leq N$, $X \neq Y$
- $1 \leq A, B \leq N$
- $1 \leq E \leq N$

Examples

Input	**Output**
7 8 9 21 33 33 18 42 22 26 1 2 1 3 2 5 3 5 3 6 4 6 4 7 6 7 P 7 T 4 2 P 7 P 5 T 1 4 P 7 T 4 7 P 2 P 6	18 21 18 18 * 26

Input	**Output**
6 5 6 10 20 30 40 50 60 1 5 1 4 3 6 2 5 4 5 P 1 P 5 P 6 T 1 6 P 1 P 4	* 10 30 30 60

Problem D

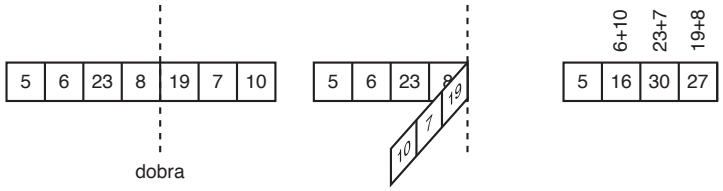
Folding Machine

File: `machine.[c|cpp|java]`

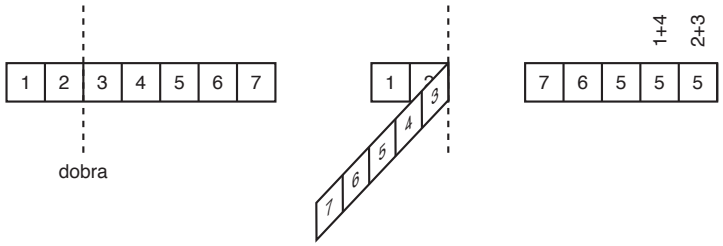
One of the main tools of a Turing machine, which allows its computing power to be bigger than other simpler models, is an infinite tape, divided in cells, where information is stored.

A Folding machine is a machine inspired by a Turing machine. In a Folding machine, the tape is finite, the data are integers and instead of having the functionality of the original Turing machine, this machine uses folding tape operations.

To perform a folding operation, the machine chooses a position between adjacent cells and folds the tape, adding the values of overlapping cells, as can seen in the figure below.



Notice that the machine can also fold the tape before the tape center, as shown in the next figure. The machine can also choose to fold at the tape start or at the tape end, actually inverting the tape.



Science of Bends Company is developing commercial versions of their Folding machine and its production have recently raised. The last lot produced, unfortunately, have some issues and some machines aren't working properly. Some additional testing is therefore needed, to avoid selling defective machines, which would denigrate the company's image.

To test these machines, a set of tests and tapes are given. For each tape, the machine returns some computation result. Therefore, the engineers responsible for testing take note of the results and can verify if they are correct. But these engineers forgot to take note of which computation was made in each test case.

To avoid re-testing all machines again, the engineers agreed that any combination of foldings is sound and accepted if, from a given input, it generates the expected output. You were hired to develop a program which, given the input and output tapes, determines whether there is a folding sequence that, starting from the input tape, generates the output tape.

Input

The input contains four lines. The first two lines refer to the input tape for the Folding machine and the last two lines refer to the output tape. The first line contains a single number, N, describing the input tape size. The second line contains N integers $v_1, \ldots, v_N$ describing the content of the input tape. The third line contains a single integer M, the output tape size; and the fourth line contains M integers $w_1, \ldots, w_M$, the content of the output tape.

Output

You program must produce a single line, containing a single character, which must be "S" if there is a folding sequence able to generate the output tape starting from the input tape, and "N" otherwise.

Restrictions

- $1 \leq M \leq N \leq 15$.
- $0 \leq v_i, w_j \leq 10^8$, for $1 \leq i \leq N$ and $1 \leq j \leq M$.

Examples

Input	**Output**
7 5 6 23 8 19 7 10 4 5 16 30 27	S

Input	**Output**
7 1 2 3 4 5 6 7 5 7 6 5 5 5	S

Input	**Output**
4 1 2 3 4 1 10	S

Input	**Output**
6 19 23 3 51 2 0 2 34 64	N

Input	**Output**
6 1 2 3 4 5 6 6 1 2 3 4 5 6	S

Input	**Output**
6 1 2 3 4 5 6 6 6 5 4 3 2 1	S

Problem E

Dangerous Dive

File: `dive.[c|cpp|java]`

The recent earthquake in Nlogonia did not affect too much the buildings in the capital, which was at the epicenter of the quake. But the scientists found that it affected the dike wall, which now has a significant structural failure in its underground part that, if not repaired quickly, can cause the collapse of the dike, with the consequent flooding the whole capital.

The repair must be done by divers, at a large depth, under extremely difficult and dangerous conditions. But since the survival of the city is at stake, its residents came out in large numbers to volunteer for this dangerous mission.

As is traditional in dangerous missions, each diver received at the start of his/her mission a small card with an identification number. At the end of their mission, the volunteers returned the nameplate, placing it in a repository.

The dike is safe again, but unfortunately it seems that some volunteers did not return from their missions. You were hired for the grueling task of, given the plates placed in the repository, determine which volunteers lost their lives to save the city.

Input

The input is composed of two lines. The first line contains two integers N and R, indicating respectively the number of volunteers that went to the mission and the number of volunteers that returned from the mission. Volunteers are identified by numbers from 1 to N. The second line contains R integers, indicating the volunteers which returned from the mission (at least one volunteer returned).

Output

Your program must produce a single line containing the identifiers of the volunteers who did not return from their missions, in ascending order of their identifications. Leave a blank space after each identifier (notice that, therefore, there must be a blank space after the last identifier in the line). If every volunteer returned, the line must contain a single character '`*`' (asterisc).

Restrictions

- $1 \leq R \leq N \leq 10^4$

Examples

Input	**Output**
`5 3` `3 1 5`	`2 4`

Input	**Output**
`6 6` `6 1 3 2 5 4`	`*`

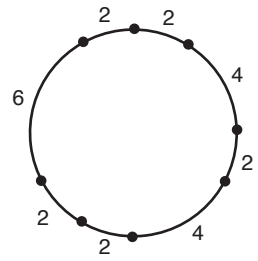
Problem F

Triangles

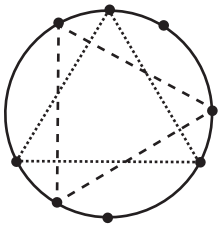
File: `triangulos.[c|cpp|java]`

You will be given N points on a circle. You must write a program to determine how many distinct equilateral triangles can be constructed using the given points as vertices.

The figure below illustrates an example: (a) shows a set of points, determined by the lengths of the circular arcs that have adjacent points as extremes; and (b) shows the two triangles which can be built with these points.



(a)



(b)

Input

The first line of the input contains an integer N, the number of points given. The second line contains N integers X_i, representing the lengths of the circular arcs between two consecutive points in the circle: for $1 \leq i \leq (N-1)$, X_i represents the length of the arc between between points i and $i+1$; X_N represents the length of the arc between points N and 1.

Output

Your program must output a single line, containing a single integer, the number of distinct equilateral triangles that can be constructed using the given points as vertices.

Restrictions

- $3 \leq N \leq 10^5$
- $1 \leq X_i \leq 10^3$, for $1 \leq i \leq N$

Examples

Input	**Output**
8 4 2 4 2 2 6 2 2	2

Input	**Output**
6 3 4 2 1 5 3	1

Problem G
Lines of Containers

File: `lines.[c|cpp|java]`

A shipment of Nlogs, the main export product from Nlogonia, is in the harbour, in containers, ready to be shipped. All containers have the same dimensions and all of them are cubes. Containers are organized in the cargo terminal in L lines and C columns, for a total of LC containers. Each container is marked with a distinct identification number, from 1 to LC.

Each one of the L container lines will be loaded in a different ship. To make it simpler when unloading at each destination country, containers from a line must be organized such that the identifiers are in sequential order. More precisely, the first line must have the containers identified from 1 to C in increasing order, line 2 must have containers numbered from $C+1$ to $2C$ (in increasing order), and so on until the last line. which will have containers numbered $(L-1)C+1$ to LC (again, in increasing order). Figure (a) shows the organization of a shipment with 5 lines and 4 columns of containers.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

(a)

13	14	15	16
5	6	7	8
9	10	11	12
1	2	3	4
17	18	19	20

(b)

13	15	14	16
5	7	6	8
9	11	10	12
1	3	2	4
17	19	18	20

(c)

A crane is able to move either a full line or a full column of containers. It cannot move other groupments or individual containers.

In night before the loading, a group of workers operated the cranes to swap shipment lines and columns as a way of protest because of low salaries. Figure (b) shows the configuration after swapping lines 1 and 4; Figure (c) shows the configuration after another swap, this time between columnns 2 and 3.

The loading must be done today, but before starting the containers must be reorganized as described previously. You must write a program which, given the information about the position of every container after the protest, determines whether you can reposition the containers in such way that every one of them is in its expected positions, using only cranes. If repositioning is possible, you must also calculate the smallest number of column and line swaps needed to do it.

Input

The first line of input contains two integers L and C which describe, respectively, the number of lines and columns of the shipment. The next L lines show the configuration of the containers after the workers protest. Each of these L lines will have C integers $X_{l,c}$, indicating the position of a container. Every integer between 1 and LC appears exactly once in the input.

Output

Your program must produce a single line, containing a single integer, the minimum number of column and line swaps needed to place the containers in their original positions. If there is no way to place

the containers in their original positions using only cranes, the line must contain only the character '*'.

Restrictions

- $1 \leq L \leq 300$
- $1 \leq C \leq 300$
- $1 \leq X_{l,c} \leq LC$

Examples

Input	**Output**
2 2 3 4 1 2	1

Input	**Output**
3 3 9 2 4 5 8 7 6 1 3	*

Input	**Output**
5 4 13 15 14 16 5 7 6 8 9 11 10 12 1 3 2 4 17 19 18 20	2

Problem H

Buses

File: `buses.[c|cpp|java]`

Programming competitions usually require infrastructure and organization on the part of those responsible. A problem that frequently must be solved is regarding transportation. While participating in a recent competition, Ricardinho watched the buses and micro-buses used in the transportation of competitors, all lined up one behind the other as competitors disembarked. The vehicles were all from the same company, although had different paintings. Ricardinho began to wonder how many ways that line could be formed using buses and minibuse from that company.

Each bus is 10 meters long, each minibus is 5 meters long. Given the total length of a line of buses and minibuses, and the number of different colors each buse or minibus may be painted, Ricardinho wants to know in how many ways such a line can be formed.

Input

The input contains a single line, containing three integers N, K and L, representing respectively the total length, in meters, of the line Ricky is considering, K indicates the number of different colors for micro-buses, and L represents the number of different colors for buses. Note that, as integers N, K and L may be very large, the use of 64 bits integers is recommended.

Output

As the number of different ways of forming the line can be very large, Ricardinho is interested in the last 6 digits of that quantity. Thus, your program should produce a single line containing exactly 6 digits, corresponding to the last digits of the solution.

Restrictions

- $5 \leq N \leq 10^{15}$ and N is multiple of 5
- $1 \leq K \leq 10^{15}$
- $1 \leq L \leq 10^{15}$

Examples

Input	**Output**
`25 5 5`	`006000`

Input	**Output**
`5 1000 1000`	`001000`

Input	**Output**
`20 17 31`	`111359`

Input	**Output**
15 9 2	000765

Problem I

Patches

File: `patches.[c|cpp|java]`

Carlos is very concerned with the environment. Whenever possible, he tries to use less polluting means of transport. He recently got a job close to home and is now using his bike to go to work.

Unfortunately, in the route between his home and his job there is a nail factory, and often some nails fall from their trucks, and end up puncturing Carlos' bike tires. Therefore he ends up having to make several patches on the tires of his bike.

To make the repairs, Carlos uses two different types of patches. Both types are as wide as a bike tire, but differ in length. As the cost of the patch is proportional to its length, Carlos is trying to find a way to save money, using the least possible length of patches to make the repairs, without cutting the patches.

The first step in repairing a tire is making a chalk mark on a position of the tire and then writing down the distances, measured clockwise, of each of the holes in relation to the chalk mark. Each hole must be completely covered by a patch. Carlão would like your help to determine, given the positions of the holes, the most economic way to make the repair.

Input

The input contains two lines. The first line contains four integers N, C, T_1 e T_2. Integer N indicates the number of holes in the tire, and C indicates the cirunference length of the tire, in centimeters. The lengths of the patches in centimeters are given by integers T_1 and T_2. The second line contains N integers F_i, representing the distance, in clockwise direction, from the chalk mark to hole i, in centimeters.

Output

Your program must print a single line, containing a single integer, the smallest total length of patches needed to make all the repairs.

Restrictions

- $1 \leq N \leq 1000$
- $1 \leq C \leq 10^6$
- $1 \leq T_1, T_2 \leq C$
- $0 \leq F_i \leq C - 1$, $1 \leq i \leq N$
- If the distance between two holes is exactly k centimeters, a patch of length k centimeters covers both holes.

Examples

Input	**Output**
`5 20 2 3` `2 5 8 11 15`	`8`

Input	**Output**
4 20 12 9 1 2 3 13	12

Problem J

Trucks

File: `trucks.[c|cpp|java]`

The Subtle Balloons Company (SBC) is the main balloon provider for programming contests; it has huge factories and warehouses, as well as an extensive truck fleet to ensure the contestants' happiness.

There are lots of competition sites in Nlogonia, and all of them hired SBC for supplying balloons for their contests. Nlogonia is an archipelago connected by several bridges. Every island of Nlogonia may have several regional sites and may also house several SBC warehouses.

When planning the routes for balloon deliveries, SBC faced a problem: for safety issues, every bridge in Nlogonia has some maximum weight limit for vehicles which cross it. And because of the great net weight of the transported merchandise, SBC operations' chief asked you to write a program to determine the maximum weight allowed to be transported between warehouses and competition sites.

Input

The first line contains three integers N, M and S which indicate, respectively, the number of islands, the number of bridges that connect the islands and the number of sites. The islands are numbered from 1 to N.

Each of the next M lines describes a bridge. The description of a bridge consists in a line with three integers A, B and W, indicating respectively the two islands connected by the bridge and the maximum weight allowed in that bridge, in tons.

All bridges are two-way roads; every pair of islands is connected by at most one bridge; and it is possible to reach every other island in the archipelago using only bridges (naturally it may be needed to pass through other islands to do so).

Each of the next S lines describe a competition site and contains two integers L and H indicating, respectively, the number of the island where this site is and the number of the island where the wharehouse which will be used to deliver the balloons to the site is.

Output

For each site in the input, in the order they were given, your program must produce a single line, containing a single integer, the biggest weight which can be transported by truck from the warehouse to the site.

Restrictions

- $2 \leq N \leq 2 \times 10^4$
- $1 \leq M \leq 10^5$
- $1 \leq S \leq 5 \times 10^4$
- $1 \leq A, B, L, H \leq N$, $A \neq B$, $L \neq H$
- $0 \leq W \leq 10^5$

Examples

Input	Output
4 5 4 1 2 9 1 3 0 2 3 8 2 4 7 3 4 4 1 4 2 1 3 1 4 3	7 9 8 7

Input	Output
4 5 2 1 2 30 2 3 20 3 4 10 4 1 40 2 4 50 1 3 1 2	20 40