In memory of
Professor Kyung-Yong Chwa
who has always been the "correct answer".

Problem Set

Please check that you have 12 problems that are spanned across 24 pages in total (including this page).

Problem A
Ant Colonies
Time Limit: 2 Seconds

A group of scientists analyzed an ant nest where several ant colonies live. They found that the ant nest is a tree structure in which each node represents the physical place where an ant colony lives, and each edge represents a tunnel connecting two ant colonies. The most interesting thing is that each colony has exactly one color and it sometimes changes its color. The color change mechanism depends on the closest pair of colonies with a certain color c among the colonies lying on the path between two given colonies A and B. The distance between two colonies is the number of tunnels of the path connecting them, that is, the number of edges of the path connecting two corresponding nodes in the tree structure.

For example, Figure A.1 (a) shows a tree structure with five ant colonies numbered 1 to 5 of colors 1, 2, 2, 2, 1, labelled in orange above the colonies, in order from colony 1 to colony 5, respectively. For color 2 and two colonies 2 and 5, the closest pair of colonies with color 2 on the path between colony 2 and colony 5 is the pair (colony 2, colony 3). But for colony 2 and colony 4, the closest pair with color 2 is the pair (colony 3, colony 4).

Suppose now that the current color 2 of colony 3 changes to color 3 as shown in Figure A.1 (b). Then there is no closest pair of colonies with color 2 on the path between colony 2 and colony 5 because only one colony has color 2. The closest pair with color 2 for colony 2 and colony 4 becomes (colony 2, colony 4).

Given colors of ant colonies, a tree structure of the ant nest, and an ordered list of update commands for the color change and query commands for the closest pair, write a program to find the closest pair of colonies with color c between the two colonies A and B for each query (A, B, c).

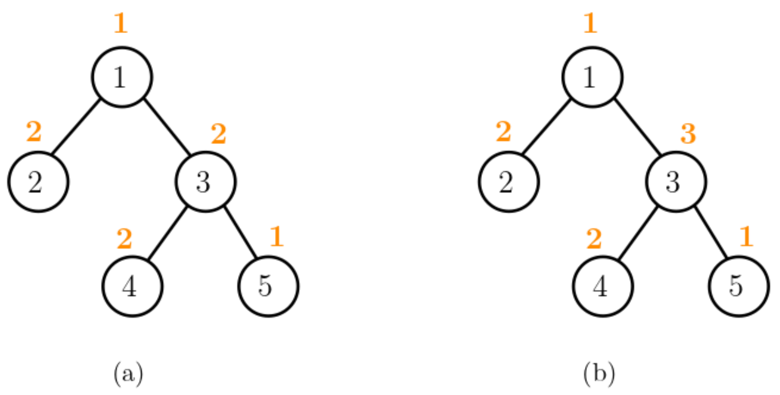


Figure A.1 An ant nest with five colonies. The numbers in orange represent colony colors.

Input

Your program is to read from standard input. The input starts with a line containing two integers, n and q ($2 \le n \le 100{,}000$, $2 \le q \le 100{,}000$), where n is the number of ant colonies and q is the number of update and query commands. Ant colonies are numbered from 1 to n, and colors are identified with integers from $\{1, 2, \dots, n\}$. The next line consists of n positive integers representing colors for ant colonies, in order from colony 1 to colony n. In the following $n - 1$ lines, the i-th line contains a pair of integers a_i, b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$) specifying the numbers of two ant colonies connected by a tunnel, which corresponds to an edge in the tree structure. In the following q lines, the i-th line has a form of (S, A, c) or (S, A, B, c), where S is a single uppercase character either ‘U’ or ‘Q’, representing the update and the query, respectively. In the case that $S =$ U, it has the form of (S, A, c) which is an update command to change (update) the current color of colony A to color c ($1 \le A, c \le n$). In the case of $S =$ Q, it has the form of (S, A, B, c) which is a query command to output the distance of the closest pair of colonies with color c on the path between colony A and colony B ($1 \le A, B, c \le n$). These commands must be executed in the order given in the input.

Output

Your program is to write to standard output. For every query (S, A, B, c) with $S =$ Q, print exactly one line containing the distance of the closest pair of colonies with color c on the path between colonies A and B under the current status of the ant nest. If there is no pair with color c between them, print `-1`.

The following shows sample input and output for two test cases. The first sample corresponds to Figure A.1.

Sample Input 1

```
5 5
1 2 2 2 1
1 2
3 1
3 4
3 5
Q 2 5 2
Q 2 4 2
U 3 3
Q 2 5 2
Q 2 4 2
```

Output for the Sample Input 1

```
2
1
-1
3
```

Sample Input 2

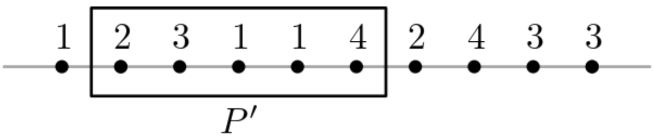
```
4 6
2 1 1 1
1 2
1 3
1 4
Q 2 3 1
Q 2 4 1
Q 3 4 1
U 1 1
Q 2 3 1
Q 2 4 1
```

Output for the Sample Input 2

```
2
2
2
1
1
```

Problem B
Double Rainbow
Time Limit: 1 Second

Let P be a set of n points on the x-axis and each of the points is colored with one of the colors $1,2,\ldots,k$. For each color i of the k colors, there is at least one point in P which is colored with i. For a set P' of consecutive points from P, if both P' and $P \setminus P'$ contain at least one point of each color, then we say that P' makes a *double rainbow*. See the below figure as an example. The set P consists of ten points and each of the points is colored by one of the colors $1, 2, 3$, and 4. The set P' of the five consecutive points contained in the rectangle makes a double rainbow.



Given a set P of points and the number k of colors as input, write a program that computes and prints out the minimum size of P' that makes a double rainbow.

Input
Your program is to read from standard input. The input starts with a line containing two integers n and k ($1 \le k \le n \le 10{,}000$), where n is the number of the points in P and k is the number of the colors. Each of the following n lines consists of an integer from 1 to k, inclusively, and the i-th line corresponds to the color of the i-th point of P from the left.

Output
Your program is to write to standard output. Print exactly one line. The line should contain the minimum size of P' that makes a double rainbow. If there is no such P', print `0`.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
10 4 1 2 3 1 1 4 2 4 3 3	5

Sample Input 2	**Output for the Sample Input 2**

`6 3`
`1`
`1`
`2`
`2`
`3`
`3`
	`0`

Problem C
Find the House
Time Limit: 1 Second

Younghee is currently on a one-dimensional road and looking for her friend Jisun's house. To know the exact position of Jisun's house, Younghee sends a message to Jisun with her current position (assume that all the positions are represented as integers). A couple of minutes later, Younghee gets a reply as a list of n triples from Jisun with an additional explanation as follows:

- For each triple (i, j, k) in the list, i is an integer which denotes the current position, j denotes the direction to move from i, represented as L (left) or R (right), and k is a positive integer which denotes the distance to move from i.
- For any two triples (i, j, k) and (i', j', k') in the list, i and i' are distinct.
- If you are currently on the position i, there always exists a triple (i, j, k) in the list (unless all the triples in the list are referred before). In this case, refer to the triple (i, j, k) and move to $i - k$ (if $j = L$) or $i + k$ (if $j = R$).
- Each of the triples in the list is referred exactly once.
- The position after referring to all the triples in the list is a position of Jisun's house.

For example, suppose Younghee is currently at the position 0 with a list of four triples – $(3, R, 4)$, $(0, L, 2)$, $(7, L, 5)$, and $(-2, R, 5)$. Then Younghee first refers to the triple $(0, L, 2)$ and move to the position $0 - 2 = -2$. After that, Younghee refers to the triples $(-2, R, 5)$, $(3, R, 4)$, and $(7, L, 5)$ in order and moves to the position 2, which is the position of Jisun's house. Given n, Younghee's current position, and a list of n triples, write a program to find Jisun's house's position.

Input
Your program is to read from standard input. The input starts with a line containing an integer n ($1 \leq n \leq 10{,}000$), where n is the number of triples in the list. In the following n lines, n triples are given where each triple is represented as three values i, j, and k, consisting of two integers i and j and one character k ($-1{,}000{,}000 \leq i \leq 1{,}000{,}000$, $j \in \{L, R\}$, and $1 \leq k \leq 2{,}000{,}000$). After n lines of triples, there is a line containing Younghee's current position as an integer between $-1{,}000{,}000$ and $1{,}000{,}000$.

Output
Your program is to write to standard output. Print exactly one line. The line should contain the position of Jisun's house.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
4 3 R 4 0 L 2 7 L 5 -2 R 5 0	2

Sample Input 2	**Output for the Sample Input 2**
5 3 L 3 -1 R 11 5 L 6 1 R 4 10 L 7 1	0

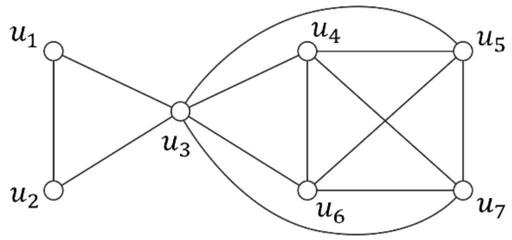
Problem D
Friendship Graphs
Time Limit: 1 Second

Given a collection of people who interact, we can define a graph whose vertices are people, with an edge between two people if and only if they are friends with one another. Such graphs are called *social networks* and are well defined on any set of people, for example, the students in a college or the residents of a small town. An entire science analyzing social networks has sprung up in recent years, because many interesting aspects of people and their behavior are best understood as properties of this *friendship graph*.

Given a friendship graph where the vertices are the students in the Problem Solving class, your job is to write a program that decomposes the students in the class into two groups, A and B, so that the following three conditions are satisfied simultaneously:

- Each student in the class belongs to exactly one group, A or B.
- Any two students in each group are friends with each other.
- The difference between the sizes of groups A and B, denoted as $\left||A|-|B|\right|$, is as small as possible.

For example, suppose we are given a friendship graph shown in the figure below. Decomposing the students into $A=\{u_1,u_2,u_3,u_6\}$ and $B=\{u_4,u_5,u_7\}$ is not possible because u_2 and u_6 are not friends. On the other hand, in the decomposition into $A=\{u_1,u_2\}$ and $B=\{u_3,u_4,u_5,u_6,u_7\}$, any two students in each group are friends with each other; however, the size difference ($|2-5|=3$) between the two groups is larger than the difference ($|3-4|=1$) in the decomposition into $A=\{u_1,u_2,u_3\}$ and $B=\{u_4,u_5,u_6,u_7\}$. The last one is an optimal decomposition we want.



Input
Your program is to read from standard input. The first line contains two integers n and m, respectively indicating the numbers of vertices and edges of a friendship graph, in which we assume $2 \le n \le 1{,}000$ and $0 \le m \le \binom{n}{2}$. The vertices are indexed from 1 to n. In the following m lines, each line contains two integers u and v which represent an edge (u,v) of the graph.

Output
Your program is to write to standard output. Print exactly one line that contains an integer. The integer should be the minimum of the size differences between two groups if the students can be decomposed into two groups satisfying the above three conditions; otherwise, the integer should be `-1`.

The following shows sample input and output for three test cases.

Sample Input 1	**Output for the Sample Input 1**
7 13 1 2 2 3 3 1 3 4 3 5 3 6 3 7 4 5 5 7 7 6 6 4 4 7 5 6	1

Sample Input 2	**Output for the Sample Input 2**
5 5 1 2 2 3 3 4 4 5 5 1	-1

Sample Input 3	**Output for the Sample Input 3**
4 4 4 3 3 2 2 1 1 4	0

Problem E
Grid Triangle
Time Limit: 0.5 Seconds

A *grid triangle* in the 3-dimensional grid system is a triangle of three integral points including the origin $(0,0,0)$ that satisfy the following property:

> There exist three different positive integers $\mathrm{X}, \mathrm{Y}, \mathrm{Z}$ such that for every pair of the three points of the triangle, you can rotate and translate the cuboid of size $\mathrm{X} \times \mathrm{Y} \times \mathrm{Z}$ in parallel with the grid system so that the pair are diagonally opposite (and so the farthest way) vertices of the cuboid.

For instance, the triangle of the three points $(0,0,0), (1,2,3), (-2,3,1)$ is a grid triangle with the cuboid of size $1 \times 2 \times 3$. More specifically, the two points $(1,2,3), (-2,3,1)$ are the diagonally opposite vertices of the cuboid $\{(x,y,z) | -2 \leq x \leq 1, 2 \leq y \leq 3, 1 \leq z \leq 3\}$ of size $3 \times 1 \times 2$; the two points $(0,0,0), (1,2,3)$ are the diagonally opposite vertices of the cuboid $\{(x,y,z) | 0 \leq x \leq 1, 0 \leq y \leq 2, 0 \leq z \leq 3\}$ of size $1 \times 2 \times 3$; and the two points $(0,0,0), (-2,3,1)$ are the diagonally opposite vertices of the cuboid $\{(x,y,z) | -2 \leq x \leq 0, 0 \leq y \leq 3, 0 \leq z \leq 1\}$ of size $2 \times 3 \times 1$. Further, all three cuboids are parallel with the grid system.

Write a program to output the number of grid triangles within a bounded 3-dimenional grid system. The grid system is bounded by three given positive integers, A, B, C, in such a way that all points of grid triangles should be within $\{(x,y,z) | -A \leq x \leq A, -B \leq y \leq B, -C \leq z \leq C\}$.

Input
Your program is to read from standard input. The input is exactly one line containing three integers, A, B, C $(1 \leq A, B, C \leq 10{,}000{,}000)$.

Output
Your program is to write to standard output. Print exactly one line. The line should contain the number of grid triangles in the 3-dimensional grid system bounded by A, B, C.

The following shows sample input and output for three test cases.

Sample Input 1	Output for the Sample Input 1
3 3 3	48

Sample Input 2	Output for the Sample Input 2
3 3 2	16

Sample Input 3	Output for the Sample Input 3
3 2 2	0

Problem F
John's Gift
Time Limit: 1.2 Seconds

Every morning, John, a storekeeper, receives n goods of distinct values and n price tags all of which have different prices. As John wants to sell as many goods as possible, he sets up a match between the goods and the price tags to minimize the maximum difference (max-difference in short) between the two pair, where different goods should match different price tags. For example, if John has two goods of values $10, 30$ and two price tags of prices $10, 20$, then the max-difference can be minimized to be 10 by matching $(10, 10)$ and $(30, 20)$. This smallest max-difference is called the matching score.

Today, Jane, a friend of John, has a birthday party and John decides to pick a birthday gift from his goods. When selecting a good, he does not want to lose too much profit, and therefore wants to select a good whose removal results in the smallest matching score for the remaining $n-1$ goods against the original n price tags. By the way, when matching $n-1$ goods, John leaves one price tag unpaired to make a proper match.

For instance, John has two goods G_1 and G_2 whose values are 10 and 30, respectively, and two price tags 10 and 20. If he picks G_1 for a gift, then a possible price for G_2 is either 10 or 20. Then the matching score is 10 when G_2 is priced at 20. On the other hand, if he picks G_2 for a gift, then the matching score is zero when G_1 is priced at 10. Therefore, in order to obtain the smallest matching score, John would select G_2 as a gift. In other words, among n goods, John can pick any single good as gift, and this defines a new matching score between the remaining $n-1$ goods against the original n price tags. Among n possible gift choices, John wants to find a good whose removal produces the smallest matching score.

Given n good values and n price tags, write a program that prints a value of a gift good that John should pick in order to produce the smallest matching score between the remaining $n-1$ goods and the n price tags. If there are two or more candidate goods to select, print the smallest value of the candidate goods.

Input
Your program is to read from standard input. The input starts with a line containing an integer n ($2 \le n \le 10^6$), where n is the number of goods and the number of price tags. The following line contains n positive and distinct integers that represent the values of n goods. The third line contains n positive and distinct integers that represent n price tags. The good values and the tag prices are no more than 10^9.

Output
Your program is to write to standard output. Print exactly one line. The line should contain the value of the good that John picks for Jane's birthday gift such that its removal produces the smallest matching score in the remaining $n-1$ goods. If there are multiple candidate goods, print the smallest value among the candidate goods.

The following shows sample input and output for three test cases.

Sample Input 1	Output for the Sample Input 1
2 10 30 10 20	30

Sample Input 2	Output for the Sample Input 2
3 20 30 40 30 20 10	40

Sample Input 3	Output for the Sample Input 3
4 24 68 51 10 20 40 50 30	68

Problem G
Logistical Warehouse 2
Time Limit: 1 Second

KOPANG is one of largest online vendors in Korea and introduced so called “early-morning delivery” for the first time. To cope with the growing demand, KOPANG plans to build new logistical warehouses. The locations of logistical warehouses must be within a certain distance from customers to keep the delivery time guaranteed by KOPANG to the customers.

The logistics network is modelled as a connected tree T. Each node of T represents a region such as a city or a province in Korea, and each edge of T represents a transportation road connecting two regions. KOPANG wants to select one or more nodes of T satisfying the *distance restriction* for the logistical warehouses. Before the selection, KOPANG first fixed a distance parameter K through sufficient research. KOPANG now wants to select the minimum number of nodes satisfying the distance restriction that the distance from every node of T to its closest selected node (warehouse) is at most K. The distance of two nodes u and v is the number of edges of the (unique) path in T that connects u and v. Note that the distance is defined as zero if $u = v$.

For example, Figure G.1 below shows a tree T with nine nodes and eight edges. For $K = 1$, if three warehouses are located at nodes 2, 5, and 8, marked with red circles as in Figure G.1 (a), then the distance of every node of T to the closest warehouse is at most one. Two warehouses are not enough to satisfy the distance restriction, so three warehouses are the minimum. For $K = 2$, three warehouses are still required; warehouses at nodes 2, 5, and 8 for $K = 1$ are the ones for $K = 2$. Of course, the locations of the minimum number of warehouses are not unique; three warehouses at nodes 4, 7, and 1 as in Figure G.1 (b) also satisfy the distance restriction for $K = 2$.

Given a connected tree T and a positive integer K, write a program to select the minimum number of nodes (warehouses) of T satisfying the distance restriction, that is, the distance of every node of T to its closest warehouse is at most K.

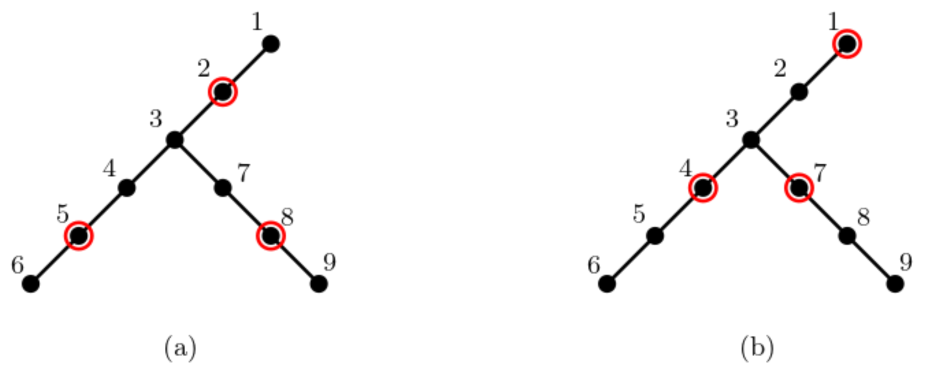


Figure G.1 The nodes marked with red circles are the ones selected for warehouses.

Input

Your program is to read from standard input. The input starts with a line containing two integers n and K ($1 \leq K \leq n \leq 10^5$), where n is the number of nodes in a connected tree and the maximum distance from each node in the tree to its closest selected node is at most K. In the following $n-1$ lines, the edge information is given; the i-th line contains two positive integers representing two indices of the end nodes of the i-th edge. The nodes are indexed from 1 to n.

Output

Your program is to write to standard output. Print exactly one line that contains the minimum number of the selected nodes for logistical warehouses satisfying the distance restriction for the given tree and the distance parameter K.

The following shows sample input and output for three test cases. The first two samples correspond to Figure G.1 (a) and (b), respectively.

Sample Input 1	Output for the Sample Input 1
9 1 2 1 7 3 3 4 4 5 6 5 7 8 3 2 8 9	3

Sample Input 2	Output for the Sample Input 2
9 2 2 1 7 3 3 4 4 5 6 5 7 8 3 2 8 9	3

Sample Input 3	Output for the Sample Input 3
9 8 2 1 7 3 3 4 4 5 6 5 7 8 3 2 8 9	1

Problem H
Postman
Time Limit: 1.5 Seconds

There is a straight road on which two types trams run. One is an east-to-west tram which moves from east to west, and the other is a west-to-east tram. For each type, several trams run regularly, so anyone can ride the tram in any direction at any time. To use the tram, you have to pay a ticket for each direction you take. In other words, to use a tram that moves from east to west, you must pay a W-ticket (west bound ticket), and conversely, to use a tram that moves from west to east, you must pay an E-ticket (east bound ticket). You can get on and off the tram at any time and place you want, and once you get on the tram, you can ride it as long as you want.

Bob, a post office worker, goes to the post office every day to deliver the mails assigned to him. He uses the tram to deliver them. Each location where mail will be delivered is represented by an x-coordinate for convenience, and the post office locates at $x = 0$.

To deliver n pieces of mail, the post office gives Bob n tram tickets. Bob uses one ticket to deliver one piece of mail. However, among the n tickets provided by the post office, the number of W-tickets is w and that of E-tickets is $e(e = n - w)$. By using the tickets he received at the post office, Bob wants not only to figure out the order in which the n pieces of mail should be delivered, but also to minimize the distance he travels using the tram.

Depending on the order in which the pieces of mail are delivered, it is divided into two types. The first type, denoted by $t = 1$, is the case that the order of mail delivery is not important. The second type, denoted by $t = 2$, is the case one specific designated piece of mail must be delivered at last and all the others can be delivered in any order.

For example, suppose that $n = 5$, $w = 4$ (the number of W-tickets), $t = 2$, and the x-coordinates of the places where the mails should be delivered are $(-20, -15, 20, 30, 10)$, and that the x-coordinate of the specific designated mail which must be finally delivered is $x = 10$. The optimal delivery route is shown in Figure H.1 and the total distance moved using trams is 90. As shown in Figure H.1, four W-tickets and one E-ticket are used and the designated mail is delivered at last.

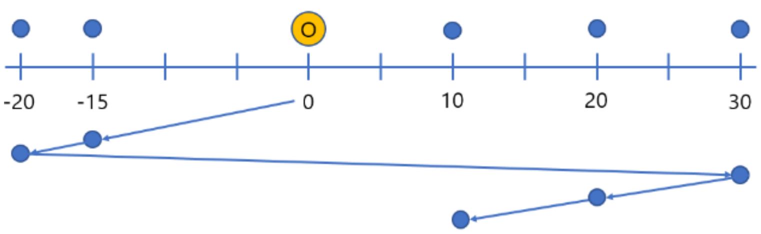


Figure H.1

Consider another example where all information is the same as above except for $t = 1$. The optimal delivery route for this case is shown in Figure H.2 and the total distance is 80. In this case, you can see that four W-tickets and one E-ticket are used as well.

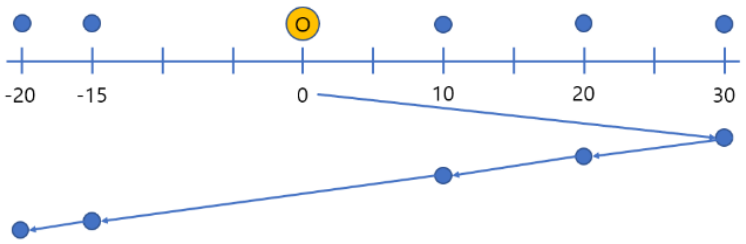


Figure H.2

Given information about the mail that Bob should deliver, write a program that finds the minimum distance he travels using trams.

Input

Your program is to read from standard input. The input starts with a line containing three integers, n, w and t ($1 \leq n \leq 3 \times 10^5$, $0 \leq w \leq n$, $1 \leq t \leq 2$), where n is the number of pieces of mail, w is the number of W-tickets, and t indicates the delivery order type as explained above. Note that the number of E-tickets is $n - w$. In the following line, n integers are given. The i-th integer x_i ($1 \leq i \leq n$, $-10^9 \leq x_i \leq 10^9$, $x_i \neq 0$) is the x-coordinate of the location where the i-th mail should be delivered. When $t = 2$, x_n denotes the x-coordinate of the specific designated mail that must be delivered at last.

You can assume no two x_i's are the same.

Output

Your program is to write to standard output. Print exactly one line. The line should contain the minimum distance Bob travels to deliver all the pieces of mail. If it is impossible for Bob to deliver them using the tickets print `-1`.

The following shows sample input and output for three test cases.

Sample Input 1	Output for the Sample Input 1
`5 4 2` `-20 -15 20 30 10`	`90`

Sample Input 2	Output for the Sample Input 2
`5 4 1` `-20 -15 20 30 10`	`80`

Sample Input 3	Output for the Sample Input 3
`7 1 2` `10 13 -30 24 50 -5 -21`	`-1`

Problem I
Security System
Time Limit: 0.8 Seconds

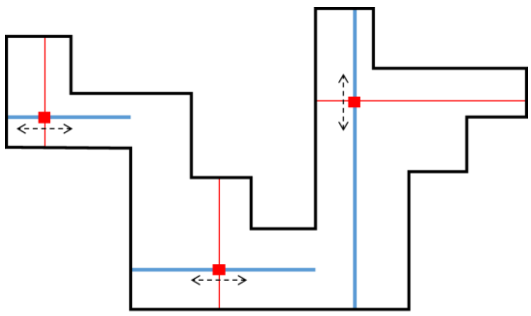
The management committee plans to introduce a new security system to monitor the museum at night time. The floor plane of the museum has a shape of a rectilinear polygon P whose edges are either horizontal or vertical. In addition, P has the x-monotone boundary, that is, the intersection of P and any vertical line is either empty or a single line segment.

The new security system is based on infrared laser beam sensors. Moving along a straight track placed inside of P, an infrared laser beam sensor unit emits the laser beam in a direction perpendicular to the track. When it detects any motion, an emergency alarm is issued immediately.

Tracks are represented as horizontal or vertical line segments. Tracks are unlimited in length. A point q in P is monitored by a sensor located at a point p on a track if $q = p$ or the following conditions are satisfied.

(i) The line segment connecting p and q does not meet the outside of P.
(ii) The track and the line segment connecting p and q are orthogonal to each other.

A polygon P is said to be completely monitored by a set T of tracks if each point inside P is monitored by a sensor on a track of T. The committee wants to know the minimum number of infrared laser beam sensor units required to completely monitor the museum. There are two things to note. The first is that the boundary of P do not intersect a track excluding its endpoints, and the second is that the tracks must not intersect each other, even at their endpoints. For example, at least 3 sensor units are required to monitor the x-monotone rectilinear polygon as shown in the figure below. In this figure, blue lines represent tracks.



Given an x-monotone rectilinear polygon, write a program to compute the minimum number of sensor units required to completely monitor the polygon.

Input
Your program is to read from standard input. The input starts with a line containing an integer, n ($4 \le n \le 100{,}000$), where n is the number of vertices of an x-monotone rectilinear simple polygon. The following n lines give the coordinates of the vertices in counterclockwise direction. Each vertex is represented by two numbers separated by a single space, which are the x-coordinate and the y-coordinate of the vertex, respectively. Each coordinate is given as an integer between $-100{,}000{,}000$ and $100{,}000{,}000$, inclusively.

Output

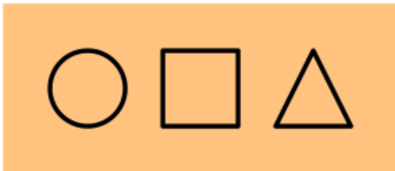
Your program is to write to standard output. Print exactly one line. The line should contain an integer representing the minimum number of sensor units required to completely monitor the given polygon.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
20 5 1 14 1 14 7 16 7 16 9 18 9 18 11 13 11 13 13 11 13 11 4 9 4 9 6 7 6 7 10 3 10 3 12 1 12 1 8 5 8	3

Sample Input 2	Output for the Sample Input 2
12 12 5 4 5 4 3 1 3 1 1 6 1 6 3 9 3 9 1 15 1 15 3 12 3	2

Problem J
Squid Game
Time Limit: 1 Second



You are one of the 456 players participating in a series of children's games with deadly penalties. Passing through a number of maze-like hallways and stairways, you have opened the gate of the next game. There are three buckets with infinite capacity, each of which contains an integral number of liters of water. The buckets are numbered from 1 to 3. The amounts of water initially contained in buckets 1, 2, and 3 are given as X, Y, and Z, respectively.

At any time, you can double the amount of one bucket by pouring into it from another one. Specifically, you can pour from a bucket of y liters into one of x $(\leq y)$ liters until the latter contains $2x$ liters and the former does $y - x$ liters. Note that x and y are always integers and $x \leq y$. See the Figure J.1.

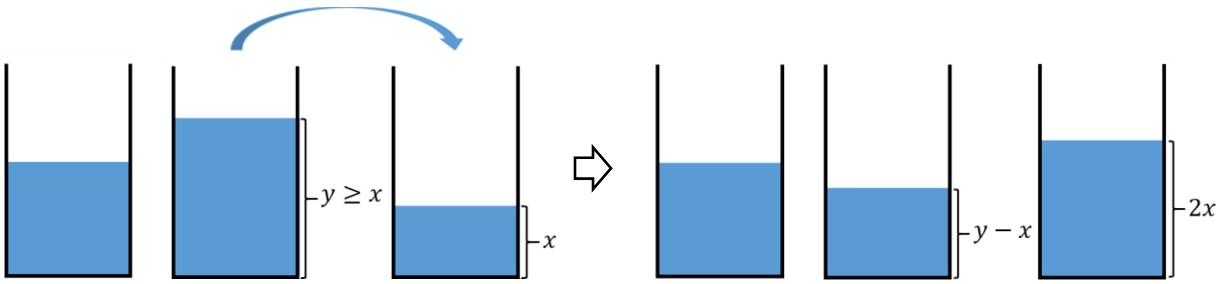


Figure J.1 A process of pouring

In order to survive, you have to empty one of the buckets in a limited number of pouring. Fortunately, it is always possible to empty one of the buckets. Given the initial amounts X, Y, and Z of water in three buckets, write a program to output a sequence of pouring until one of the buckets is empty for the first time.

Input
Your program is to read from standard input. The input starts with a line containing three integers X, Y, and Z $(1 \leq X \leq Y \leq Z \leq 10^9)$, representing the initial amounts of water in buckets 1, 2, and 3, respectively.

Output
Your program is to write to standard output. The first line should contain the number m of pouring until one of the buckets is empty for the first time. The number m should be no more than 1,000. Each of the following m lines contains two integers A and B $(1 \leq A \neq B \leq 3)$, which means you pour from bucket A into bucket B in a process of pouring. You should guarantee that one of buckets is empty for the first time after the m pouring. If there are several ways to empty one of the buckets, then print one of them.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
1 2 3	2 3 2 3 1

Sample Input 2	Output for the Sample Input 2
1 4 6	3 2 1 3 1 1 3

Problem K
Stock Price Prediction
Time Limit: 1 Second

Mr. Kim is a stock market analyst. Recently, he found something interesting while looking at the stock charts of several companies. Most of the stocks that rose for four consecutive days fell the next day. Also, the stock price that fell on the fifth day was often located between the price of the second and third days of the stock price during the four days of the uptrend. For example, Company A's stock price was 500 won, 560 won, 600 won, and 680 won for four consecutive days, and the fifth day's stock price of Company A was 580 won. Also, Company B's stock price was 1,000 won, 1,200 won, 1,400 won, and 1,700 won for four consecutive days, and the fifth day's stock price of Company B was 1,350 won.

Mr. Kim thinks that if he can find a part in the previous stock price sequence that matches the price movement pattern over the last few days, he will be able to predict the stock price for the next day quite accurately. He also thinks that the relative ranks in a stock price sequence are more important than the actual prices because if the relative ranks of two stock price sequences are the same, their patterns in charts look similar. In the above example, the stock price sequence of Company A for five consecutive days, 500 won, 560 won, 600 won, 680 won, 580 won, can be represented as $(1, 2, 4, 5, 3)$ because 500 is the smallest among the five numbers, 550 is the second smallest, 600 is the fourth, and so on. Moreover, the stock price of Company B for five consecutive days, 1,000 won, 1,200 won, 1,400 won, 1,700 won, 1,350 won, can also be represented as $(1, 2, 4, 5, 3)$ due to the same reason. Their relative ranks are the same and their charts of five consecutive days look very similar as shown in Figure K.1.

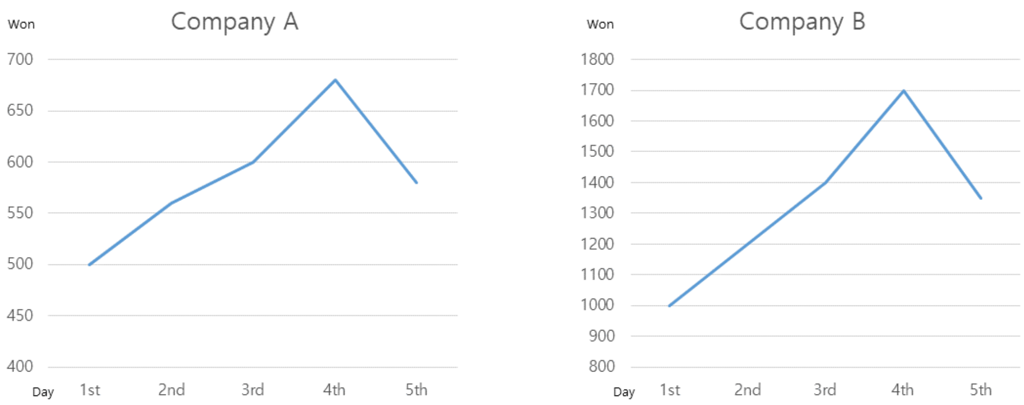


Figure K.1 Charts of Company A and B for five consecutive days

Mr. Kim decided to consider two sequences as a match if all the relative ranks of same positions of two sequences are the same. Mr. Kim formally defined *R-match* of two sequences of same length (number of integers) as follows: Two sequences of integers $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_m)$ of the same length are an *R-match* if and only if for each i $(1 \le i \le m)$, x_i's rank in x and y_i's rank in y are the same. Next, he defined the *R-pattern matching problem* as follows: Given two sequences of integers x of length m and y of length n $(m \le n)$, find every position i of y such that x and $(y_i, \ldots, y_{i+m-1})$ are an R-match. For example, when $x = (33,40,22,40,41,28)$ and $y = (10,20,16,27,32,12,32,33,20,25,15,25,31,17)$, x and $(y_4, \ldots, y_9)$ are an R-match. Also, x and $(y_9, \ldots, y_{14})$ are an R-match.

Given two sequences of integers x of length m and y of length n ($m \leq n$), write a program to solve the R-pattern matching problem for x and y.

Input

Your program is to read from standard input. The input starts with a line containing two integers, m and n ($1 \leq m \leq 10{,}000$, $1 \leq n \leq 1{,}000{,}000$, $m \leq n$), where m is the length of x, and n is the length of y. In the second line, the m integers in x are given in turn. In the third line, the n integers in y are given in turn. Each integer in x and y ranges from 1 to 10^9.

Output

Your program is to write to standard output. Print exactly one line. The line should contain every position i of y such that x and $(y_i, \dots, y_{i+m-1})$ are an R-match. Each position must appear in increasing order. If there is no such position, print `0`.

The following shows sample input and output for three test cases.

Sample Input 1	Output for the Sample Input 1
5 12 500 560 600 680 580 30 25 40 60 70 90 65 30 35 50 55 40	3 8

Sample Input 2	Output for the Sample Input 2
5 15 1000 1200 1400 1700 1350 1 2 3 4 5 6 7 8 7 6 5 4 3 2 1	0

Sample Input 3	Output for the Sample Input 3
6 14 33 40 22 40 41 28 10 20 16 27 32 12 32 33 20 25 15 25 31 17	4 9

Problem L
Trio
Time Limit: 2 Seconds

Let A be any set of n natural numbers whose decimal representations consist of exactly four digits without 0 in any decimal place.

A *trio* is a set of three numbers $\{a, b, c\}$ chosen from A such that the following conditions are fulfilled simultaneously:

- The ones decimals of three numbers a, b, c are either all equal or all distinct.
- The tens decimals of three numbers a, b, c are either all equal or all distinct.
- The hundreds decimals of three numbers a, b, c are either all equal or all distinct.
- The thousands decimals of three numbers a, b, c are either all equal or all distinct.

For examples, $\{1425, 1113, 1354\}$ is a trio if the three numbers are members of A because the ones decimals of the three numbers are all distinct, their tens decimals are all distinct, their hundreds decimals are all distinct, and their thousands decimals are all equal. The set $\{1425, 1113, 5436\}$, however, is not a trio, even if A contains those three numbers.

Given a set A as input, write a program that computes and prints out the number of different trios.

Input
Your program is to read from standard input. The input starts with a line consisting of a single integer n ($1 \leq n \leq 2{,}000$) that represents the number of members in A. Each of the following n lines consists of a positive integer in decimal form that consists of exactly four digits without 0 in any decimal place. These n numbers are supposed to be all distinct and the members of the input set A.

Output
Your program is to write to standard output. Print exactly one line. The line should consists of a single integer that represents the number of different trios for the input set A.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
6 1234 1235 1244 1233 7133 8133	1

Sample Input 2	Output for the Sample Input 2
9 1234 5678 9123 4567 8912 3456 7891 2345 6789	84