Problem A: Maria's Midpoint Mystery

Filename: `midpoint`

Timelimit: `2 seconds`

One day while bored, Maria decided to take a piece of paper and place ***N***+1 dots in what she thinks were random integer positions along a straight line on the paper. Feeling a little fiddly, she also marked the midpoints between each adjacent dot on the line.

The next day she woke up to find that the dots have been erased. She wants to restore the dots on the paper at integer coordinates, but has no clue where to actually put the dots. Really any configuration of dots will do, so long as 1) the midpoints of the previous paper are also midpoints of the new dots, 2) no dot shares a coordinate, and 3) all the new dots lie on integer coordinates. But, as Maria knows someone messed with her paper, she also knows some of the midpoints might have also been tampered with, so she wonders if it's even possible to reconstruct the dots on integer coordinates.

Input

The first line of input contains a single positive integer ***n*** ($2 \le n \le 10^5$), representing the number of midpoints. The next line contains ***n*** space separated integers; the *i*th integer on this line denotes the signed distance the midpoint between point i and point i+1 is from the center of the paper. These integers will be unique, ordered from least to greatest, and within in the range [-10^{18}, 10^{18}], with integers less than 0 representing points to the left of the origin of the line and integers greater than 0 representing points to the right of the origin of the line.

Output

If it is impossible to draw dots on the lines at integer coordinates such that the dots match up with the paper, output `-1`. Otherwise, output two integers, representing the leftmost and rightmost integer positions where the first dot might be (it's fairly easy to tell the positions of the rest of the dots from this one coordinate).

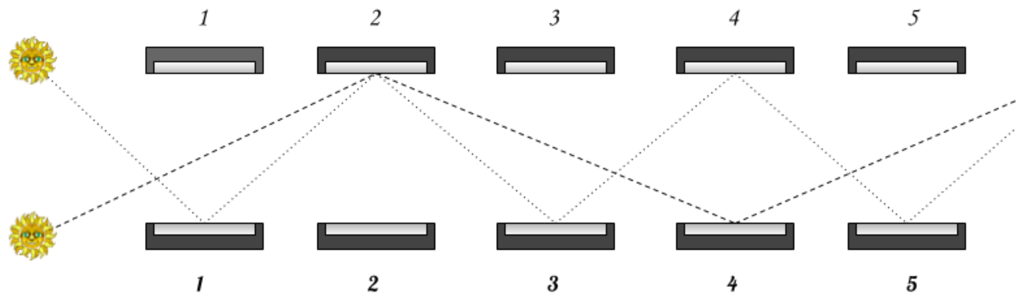
Samples

Input	Output
5 2 6 14 20 25	1 1
4 -3 3 7 13	-8 -6
7 -7 -2 2 8 12 20 27	-1

Problem B: Hall of Mirrors

Filename: `mirror`

Timelimit: `1 second`



While working their way through the Earth Temple, Link and Medli came across a room filled with two rows of mirrors. The room has two rays of sun shining down from the ceiling. Link can use his mirror shield and Medli her golden harp to direct light to the mirrors. Some mirrors have glyphs on them that give off a blue glow while light is reflected off them. Link suspects all glyphs need to be glowing to solve the puzzle of room, so they want to know the minimum number of rays necessary to light up all the glyphs on the mirrors. There are n mirrors in both the top and bottom rows each row is separately numbered from 1 to n. The center of the ith mirror on the top row is at location (i, 1) while the center of the ith mirror on the bottom row is at (i,0). The two sunspots are at locations (0,1) and (0, 0). Multiple rays of light can be reflected from the same sunspot. Rays can light up glyphs regardless of the number of times it has been reflected off a mirror. Since Link and Medli use Z-targeting, they can only aim for the centers of mirrors.

Input

The first line of input will contain three space separated integers, n ($1 \le n \le 10^5$) the number mirrors per row. The next line will first contain an integer t ($0 \le t \le n$), which is followed t integers t_i ($1 \le t_i \le n$), the mirrors on the top row that hold glyphs. The next line contains an integer b ($0 \le b \le n$) followed by b integers b_i ($1 \le b_i \le n$) the mirrors of the bottom row that hold glyphs. Glyphs will be given in increasing order.

Output

On a line by itself output a single integer, the minimum number of rays that need to be reflected to light up all the glyphs.

Samples

Input	Output
5 1 4 4 1 3 4 5	2
5 2 3 5 0	1
30 0 3 8 16 24	1
1 0 0	0

Problem C: N Queens

Filename: `nqueens`
Time limit: `2 seconds`

We wish to solve the N Queens problem - how to place n queens on an n by n chessboard so that no two are attacking each other. However, we wish to only find the lexicographical first solution. Let $\{p_1, p_2, p_3, \ldots, p_N\}$ be a permutation of the set {1, 2, 3, …, N}. We let each permutation describe a placement of queens as follows: p_i represents the column in which the queen on row i is placed. Thus, the permutation {2, 4, 1, 3} represents queens in row 1 column 2, row 2 column 4, row 3 column 1 and row 4 column 3, as shown below:

	Q		
			Q
Q			
		Q	

This is the first lexicographical solution for N = 4, since none of the queens shown above share the same row, column or diagonal. (Note: To compare two solutions in lexicographical ordering, find the first corresponding number that differs. The one that comes first is the one that has a lower number for the first differing number. Thus, 2, 4, 1, 3 comes before 3, 1, 2, 4, but after 2, 3, 4, 1.)

Input

The input consists of a single positive integer, ***n*** ($4 \le n \le 15$), representing that the chess board for the input case is size ***n*** x ***n***.

Output

Output on a single line ***n*** space separated integers representing the first lexicographical solution to the ***n*** queens problem as described above.

Samples

Input	Output
4	2 4 1 3
5	1 3 5 2 4

Problem D: Presidential Security

Filename: `security`
Time limit: `1 second`

After hearing that you had attended SI@UCF Programming Competition Camp, the White House was so impressed, that they hired you to help lay out the communication network when the President traveled. The president only likes staying in hotels that have all of their rooms on each floor on a row. The rooms are numbered left to right, and no floor has more than 100 rooms, so the last two digits of each room uniquely identify where the room is within a given floor. All digits beyond the last two digits signify the floor of the room. For example, room 6287 is the 88^{th} room from the left (starting from 0) on the 62nd floor. Unfortunately, the president doesn't travel light. His party may end up taking several rooms at a hotel, and often times, to avoid suspicion, the team stays scattered throughout the hotel.

For example, consider a hotel with 4 floors and 8 rooms per floor (shown below) and the President's party staying in rooms 101, 203, 207, 402 and 406. The President would like you to set up wired communication that links all five rooms, either directly or indirectly, in the shortest time possible.

400	**401**	**402**	**403**	**404**	**405**	**406**	**407**
300	**301**	**302**	**303**	**304**	**305**	**306**	**307**
200	**201**	**202**	**203**	**204**	**205**	**206**	**207**
100	**101**	**102**	**103**	**104**	**105**	**106**	**107**

For all hotels with this layout, you've discovered that you can wire any two rooms directly and that the amount of time (in minutes) it will take you to lay a single wire between two rooms is ***a(dx) + b(dy)***, where ***a*** and ***b*** are given constants (different for each hotel) and ***dx*** represents how many rooms separate the two rooms on the x axis, which runs horizontal to the ground, and ***dy*** represents how many rooms separate the two rooms on the y axis, which runs vertically. For example, if ***a*** = 2 and ***b*** = 9, then we could connect the following sets of rooms for the hotel above:

101 and 203, time = 2(3 - 1) + 9(2 - 1) = 13
203 and 402, time = 2(3 - 2) + 9(4 - 2) = 20
203 and 207, time = 2(7 - 3) + 9(2 - 2) = 8
402 and 406, time = 2(6 - 2) + 9(4 - 4) = 8

in 13 + 20 + 8 + 8 = 49 minutes. This arrangement allows all pairs of room to communicate and all other alternate arrangements would take you equal or more time to set up as this one.

Write a program to calculate the minimum time it'll take you to set up a network the president desires for various hotels.

Input

The first line of input will contain three space separated positive integers, ***n***, (***n*** ≤ 100), representing the number of rooms the President's party is taking in that hotel, ***a***, (***a*** ≤ 100) and ***b*** (***b*** ≤ 100), representing the two given constants for the hotel in the input case.

The next ***n*** lines of the input contain one positive integer, each, representing one of the room numbers in the President's party. Each of these integers is guaranteed to be distinct and in between 100 and 9999, inclusive.

Output

On a line by itself, output the minimum number of minutes it will take you to set up the network for the president.

Samples

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>5 2 9
101
203
207
402
406</td><td>49</td></tr>
<tr><td>3 1 100
100
199
200</td><td>199</td></tr>
</table>

Problem E: Ultimate Frisbee

Filename: `frisbee`
Time limit: `2 seconds`

After a long day of contests, it's time for a friendly game of ultimate frisbee between campers. Throughout the competition days, several rivalries have formed between campers. The campers want the rivalries to continue even through frisbee.

Your task is to divide the campers into two teams as evenly as possible (if possible) where no two campers that are rivals are on the same team.

Input

The first line contains two integers n and m ($1 \le n \le 2000$, $1 \le m \le 10^5$), the number of campers and the number of rivalries, respectively.

The next m lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$), representing a rivalry between campers a_i and b_i.

Output

Output two integers a and b, where $a \le b$, representing the sizes of the two teams where $b-a$ is minimized. If such a division is not possible, print "Impossible" without quotes.

Samples

Input	Output
3 3 1 2 2 3 1 3	Impossible
10 8 1 2 1 3 1 4 1 5 1 6 1 7 8 9 9 10	3 7
7 8 1 2 1 3 3 4 2 4 4 5 4 6 5 7 6 7	3 4

Problem F: Text Messages

Filename: `messages`
Time limit: `3 seconds`

Arup has a favorite graph, G, that has N nodes and M edges. Matt has a tree, T. Matt's tree can be formed by removing a subset of edges from Arup's graph. Arup knows that Matt's tree is formed by a subset of edges from his graph, but he does not know what subset this is. Arup is going to send Matt some lists of edges such that Matt can determine exactly what G is. Formally, if Matt adds all edges that Arup sends him to T (that aren't already in T), T will now be identical to G. Arup plans on texting Matt these lists, but due to his odd data plan he can only send a special type of list. For every edge pair of adjacent edges in a list, the edges must share a node. Additionally, the first edge in every list must share a node with the last edge in the list. To make matters more complicated, Matt will get bored if an edge shows up more than once in Arup's messages (including his previous lists). Please help Arup find the minimum number of lists he can send.

Input

The first line of input contains two space separated positive integers n($n \le 10^5$) and m($m \le 2\times10^5$), the number of nodes in G and the number of edges in G, respectively. The following m lines will contain information about each edge. The i^{th} of these lines will contain two space separated integers: a_i and b_i ($1 \le a_i < b_i \le N$), the two nodes that the i^{th} edge connects. No two edges will be the same.

Output

Output a single integer, the minimum number of lists Arup can send that allow Matt to determine exactly what G is. If this cannot be done without boring Matt, print -1.

Samples

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

Problem G: Ultimate Consulting

Filename: `uconsulting`
Time limit: `3 seconds`

Brian Mulch is too smart for his own good. Companies hire him because of his reputation to give business advice, but they never believe him when he tells them that their entire business strategy is flawed. Convincing them is such a chore. Not only will employees not believe him unless their direct superiors agree with him, but he was to convince everyone personally. You would think that if all your superiors were convinced that would be convincing enough, but it appears that everyone else has trouble communicating even the simplest of ideas. Because he is so perceptive, Mr. Mulch knows exactly how much mental energy is necessary to convince someone and how much mental energy will be drained in the process. For any given firm Brian would like to know the minimum starting mental energy required to convince everyone. There may be multiple possible orderings in which to convince people and some may be more efficient than others.

Input

Each test case starts with line containing a single positive integer n ($n \leq 10^5$) the number of employees at a firm. Proceeding this are n lines each with three integers s_i r_i d_i describing an employee. s_i ($0 \leq s_i < n, s_i < i$) is the supervisor of employee i or -1 if this employee is the CEO. Employees are numbered starting at 0 in the order of the input and the CEO is always the employee numbered 0. r_i is the mental energy necessary to convince employee i, and d_i is the mental energy drained in the process ($0 \leq d_i \leq r_i \leq 10^3$).

Output

For each test case output a single integer, the minimum mental energy needed to convince everyone.

Samples

Input	Output
4 -1 2 0 0 2 1 0 5 2 1 10 1	11
3 -1 4 2 0 6 2 0 5 3	9

Problem H: Ultimate Ultimate Frisbee

Filename: `ufrisbee`
Time limit: `2 seconds`

After a long day of contests, it's time for a friendly game of ultimate frisbee between campers. Throughout the competition days, several rivalries have formed between campers. The campers want the rivalries the continue even through frisbee.

Your task is to divide the campers into two teams as evenly as possible (if possible) where no two campers that are rivals are on the same team.

Input

The first line contains two integers n and m ($1 \le n \le 2 \cdot 10^5$, $1 \le m \le 2 \cdot 10^5$), the number of campers and the number of rivalries, respectively.

The next m lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \neq b_i$), representing a rivalry between campers a_i and b_i.

Output

Output two integers a and b, where $a \le b$, representing the sizes of the two teams where b-a is minimized. If such a division is not possible, print "Impossible" without quotes.

Samples

Input	Output
3 3 1 2 2 3 1 3	Impossible
10 8 1 2 1 3 1 4 1 5 1 6 1 7 8 9 9 10	3 7