Fire

In the old Baltic religion, it is important to have a holy fire burning. A priest called *krivis* is responsible for protecting it from extinguishing. He has many trustworthy helpers called *vaidilutės*, and wants to create a schedule for them to stoke and protect the fire. He has to ensure that the fire is always maintained by some *vaidilutė*.

Krivis has his own time measurement system, where each day has M minutes. There are N *vaidilutės* in his village. The i-th *vaidilutė*'s possible work time are described by two integers s_i and e_i. The number s_i is her own earliest time in the day when she may start working, and the number e_i is the latest time of the day when she needs to finish working. Time is counted in minutes from the start of the day. Note that when $s_i > e_i$, the *vaidilutė* is willing to work overnight.

Krivis asked you to choose some *vaidilutės* and arrange shifts for them. A chosen *vaidilutė* must start her shift not earlier than time s_i, and end her shift not later than e_i. A single shift is always shorter than the whole day. The chosen *vaidilutės* will repeat their shifts everyday.

Handing things over from one *vaidilutė* to the next increases the risk of the fire extinguishing. Because of this, you want to minimize the number of times this happens during the day and will arrange a schedule where the smallest possible number of *vaidilutės* is needed.

Task

Calculate the minimum number of *vaidilutės* that you need to choose, such that the holy fire is maintained at all times.

Input

The first line contains two integers N and M – the number of *vaidilutės* available and the length of the day in minutes.

Then N lines follow. The i-th of them contains two integers s_i and e_i – the earliest starting time and the latest finishing time of the i-th *vaidilutė*.

Output

Output one integer – the minimum number of *vaidilutės* you need to choose. If it is impossible to choose the *vaidilutės* according to the requirements, output -1.

Examples

Input	Output	Explanation
4 100 10 30 30 70 20 40 60 20	3	You can choose the 1-st, 2-nd and 4-th *vaidilutės* and arrange their shifts as follows: • 1-st *vaidilutė* works from the 10-th minute until the 30-th minute. • 2-nd *vaidilutė* works from the 30-th minute until the 70-th minute. • 4-th *vaidilutė* works from the 70-th minute until the 10-th minute on the following day.
1 100 30 40	-1	It is impossible to make a schedule since there is only one *vaidilutė* and she cannot work the whole day.

Constraints

- $1 \leq N \leq 2 \cdot 10^5$
- $2 \leq M \leq 10^9$
- $0 \leq s_i, e_i < M$ (for all $1 \leq i \leq N$)
- $s_i \neq e_i$ (for all $1 \leq i \leq N$)

Subtasks

No.	Points	Additional constraints
1	14	$N \leq 20$.
2	17	$N \leq 300$.
3	9	$N \leq 5\,000$.
4	13	For all *vaidilutės*, $s_i < e_i$ or $e_i = 0$.
5	21	For each *vaidilutė*, the time interval from time s_i until time e_i has the same length.
6	26	No additional constraints.

Tiles

Soon after converting to Christianity, it is believed that the first and the only Lithuanian King Mindaugas ordered the construction of the Vilnius Cathedral. The construction is almost completed, except that the floor has to be covered with ceramic ornamented glazed tiles.

The floor of the Vilnius Cathedral is a polygon in a 2D plane with a Cartesian coordinate system. The polygon has N distinct vertices, numbered from 1 to N. For each i such that $1 \leq i \leq N$, vertex i is located at point $(X[i], Y[i])$, where $X[i]$ and $Y[i]$ are nonnegative integers. There is an edge connecting vertex i and vertex $i+1$ (for each i such that $1 \leq i \leq N-1$), as well as an edge connecting vertex N and vertex 1. The vertices are listed in either clockwise or counterclockwise order.

The cathedral is an **axis-aligned** polygon, which means that each of the edges is parallel to either the x-axis or the y-axis. Moreover, the cathedral is a **simple** polygon, that is:

- exactly two edges meet at each vertex;
- any pair of edges can only meet at a vertex.

The builders of the cathedral have infinitely many pieces of tiles. Each piece is a square with side length equal to 2. The builders would like to cover a big part of the cathedral with these pieces. Specifically, the builders want to pick some vertical line and cover the part of the cathedral to the left of the line. For any integer k, let L_k denote the vertical line consisting of points with x-coordinate equal to k. A covering of the part of the cathedral to the left of L_k is a placement of some number of pieces in the plane such that:

- each point which lies in the interior of the polygon and has x-coordinate less than k is covered by some piece;
- no point which lies outside of the polygon or has x-coordinate greater than k is covered by some piece;
- the interiors of the pieces do not overlap.

The minimum x-coordinate of any vertex in the cathedral is 0. Let M denote the maximum x-coordinate of any vertex in the cathedral.

Task

Help the builders of the Vilnius Cathedral by determining the largest integer k, such that $k \leq M$, and there exists a covering of the part of the cathedral to the left of L_k. Note that by definition,

there exists a covering of the part of the cathedral to the left of L_0 (which uses 0 pieces).

Input

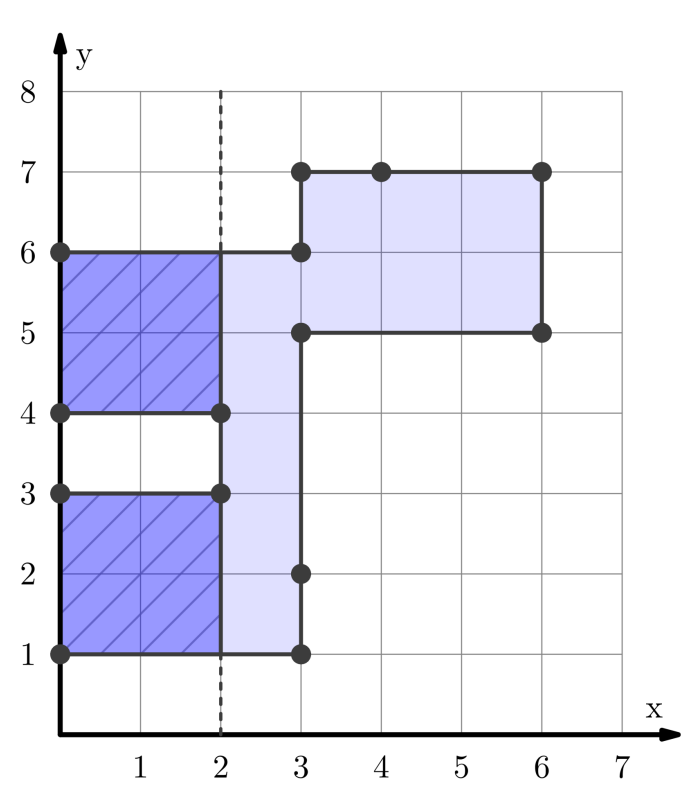
The first line of the input contains two integers N and M – the number of vertices and the maximum x-coordinate of any vertex.

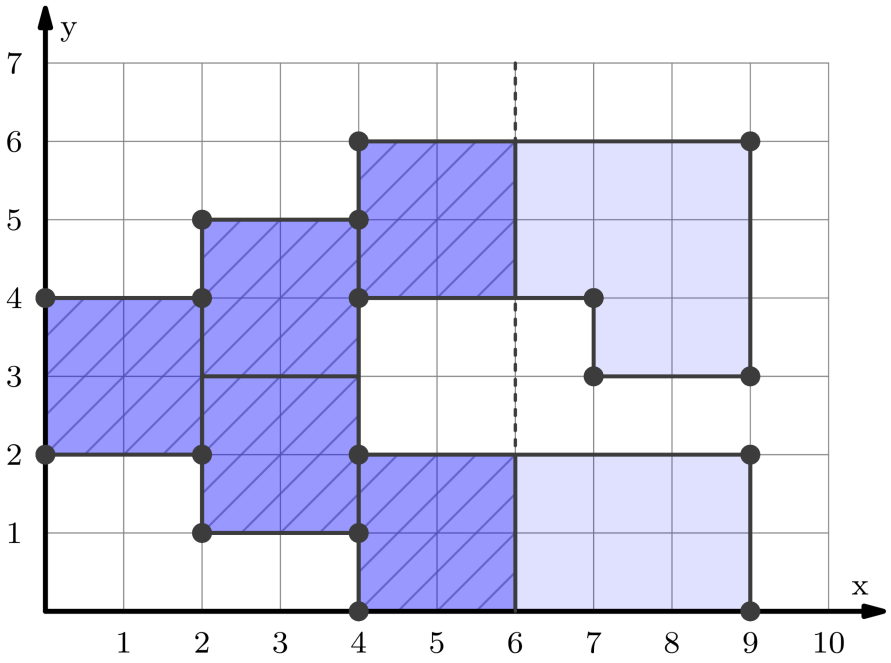
Then, N lines follow. The i-th of them contains two integer numbers x_i and y_i – the coordinates of i-th vertex. The vertices are listed in either clockwise or counterclockwise order.

Output

Your program should output the maximum k, such that $k \leq M$ and there exists a covering of the part of the cathedral to the left of L_k.

Examples

Input	Output	Explanation
14 6 0 1 0 3 2 3 2 4 0 4 0 6 3 6 3 7 4 7 6 7 6 5 3 5 3 2 3 1	2	The following picture shows the part of the cathedral to the left of line L_k for $k = 2$:  There is a covering of the part of the cathedral to the left of L_2. The covering uses two pieces. For any $k > 2$, there is no covering of the part of the cathedral to the left of L_k.

4 3 0 0 0 3 3 3 3 0	0	There is no positive value of k such that the part of the cathedral to the left of L_k could be covered with tiles.
18 9 0 2 2 2 2 1 4 1 4 0 9 0 9 2 4 2 4 4 7 4 7 3 9 3 9 6 4 6 4 5 2 5 2 4 0 4	6	As illustrated below, it is possible to cover the part of the cathedral to the left of line L_6:  For each $k > 6$, there is no covering of the part of the cathedral to the left of L_k.

Constraints

- $4 \leq N \leq 2 \cdot 10^5$
- $1 \leq M \leq 10^9$
- $0 \leq y_i \leq 10^9$ (for each $1 \leq i \leq N$)
- The cathedral forms an axis-aligned simple polygon.
- The minimum of $x_1, x_2, ..., x_N$ is 0, and the maximum of $x_1, x_2, ..., x_N$ is M.

Subtasks

No.	Points	Additional constraints
1	4	$N = 4$.
2	9	$N \leq 6$.
3	11	$x_N = 0, y_N = 0$, $x_i \leq x_{i+1}, y_i \geq y_{i+1}$ (for each i such that $1 \leq i \leq N - 2$).
4	19	$M \leq 1000$ and all $y_i \leq 1000$.
5	22	All values of y_i are even.
6	25	All values of x_i are even.
7	10	No additional constraints.

Flooding Wall

It's the 14th century and construction of the Trakai Island Castle is to begin soon. The first task on the chief architect's list is to plan the construction of the main castle wall.

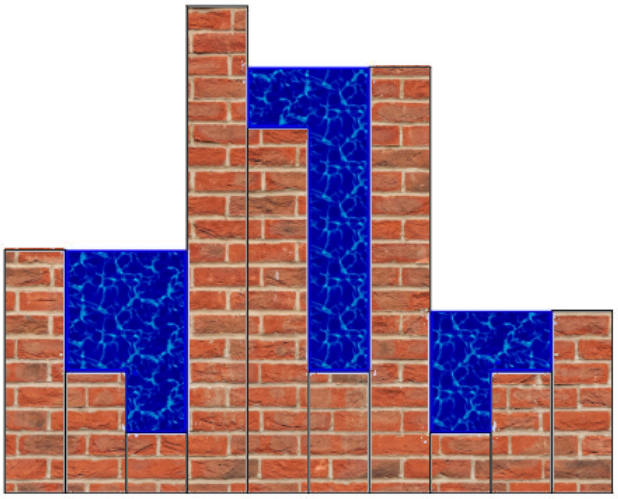
Building a wall that can protect the castle from any possible attack is quite tricky. To ensure the safety of the castle garrison, the chief architect has already narrowed the design space somewhat.

Since attacks from the middle of the lake aren't as likely as attacks from the nearby shore, the wall does not need to form a closed loop. Instead, it will be in the shape of a straight line, and consist of N segments arranged from one end to the other and numbered 1 to N. What still remains is picking the height of each segment.

The chief architect has already picked two possible heights for each segment. He decided that the height of the i-th segment will be either a_i or b_i. Thus, 2^N possibilities remain.

Having the castle on a small island in a lake has its difficulties. During stormy weather, the castle can get flooded. In such cases, water collects above wall segments if there are higher segments to each side of them, preventing the water from draining.

For a particular choice of the segments' heights, we are interested in the amount of water that will collect on the wall after a heavy storm. This is illustrated in the following figure, where the segments' heights from left to right are $4, 2, 1, 8, 6, 2, 7, 1, 2, 3$ and the level of water at each position is $4, 4, 4, 8, 7, 7, 7, 3, 3, 3$.



Formally, for every $i = 1, 2, ..., N$, the level of water at position i is at least h if and only if there exist integers l and r such that $l \leq i$ and $i \leq r$ and the segment heights at positions l and r are at least h. In particular, the level of water at positions 1 and N is always equal to the heights of the corresponding segments, and the level of water at any position is always at least as large as the height of the corresponding segment. The amount of water that collects at position i is equal to the difference between the level of water and the height of the segment. The total amount of water collected is just the sum of collected water at positions $1, 2, ..., N$.

Task

Your task is to compute the sum, over all 2^N possible walls, of the total amount of collected water. You should output the answer modulo $10^9 + 7$.

Input

The first line of the input contains one integer number N.

The second line of the input contains N integers $a_1, a_2, ..., a_N$.

The third line of the input contains N integers $b_1, b_2, ..., b_N$.

Output

Your program should output a single integer, the sum of the total amount of water collected over all 2^N possible walls modulo $10^9 + 7$.

Examples

Input	Output	Explanation
4 1 1 1 1 2 2 2 2	6	There is a single possible wall where two units of water are collected: • $2\,1\,1\,2$ and four possible walls where one unit of water is collected: • $1\,2\,1\,2$, • $2\,1\,2\,1$, • $2\,1\,2\,2$, • $2\,2\,1\,2$.

10 1 2 3 4 5 6 7 8 9 10 10 9 8 7 6 5 4 3 2 1	21116	

Constraints

$1 \leq N \leq 5 \cdot 10^5$.

$1 \leq a_i, b_i \leq 10^9$ and $a_i \neq b_i$ (for all $1 \leq i \leq N$).

Subtasks

No.	Points	Additional constraints
1	8	$N \leq 20$.
2	17	$N \leq 100$ and for all segments, $a_i, b_i \leq 1\,000$.
3	19	$N \leq 10\,000$ and for all segments, $a_i, b_i \leq 1\,000$.
4	14	$N \leq 10\,000$.
5	12	For all segments, $a_i, b_i \leq 2$.
6	30	No additional constraints.