INTERNATIONAL TOURNAMENT IN INFORMATICS
26 November, 2016, Shumen, Bulgaria
Junior Group

Task B1. PATHS IN MULTIGRAPH

Given is an undirected multigraph G without loops.

Write a program **`disconnect`**, which determines the minimum number of edges that must be removed from *G*, so that the resulting multigraph to be disconnected.

Input

The first line of the standard input contains an integer n – the number of vertices in *G*. The vertices of the multigraph *G* are numbered from 1 to n. The second line of the standard input contains an integer m – the number of edges in *G*. The next m lines of the standard input contain the endpoints u and v of the consecutive edge in multigraph *G*.

Output

The only line of the standard output has to contain one integer – the minimum number of edges that must be removed from *G*, so that the resulting multigraph to be disconnected.

Constraints

$2 \leq n \leq 100$

$0 \leq m \leq 3000$

Examples

Input	Output

`2`
`3`
`1 2`
`1 2`
`1 2`
	`3`

Input	**Output**

`3`
`2`
`1 2`
`2 3`
	`1`

Input	**Output**

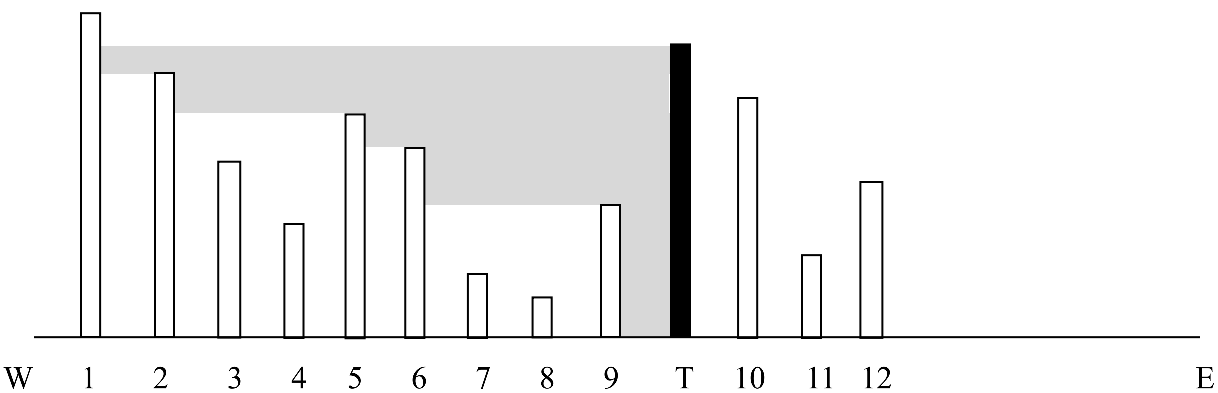
`3`
`1`
`1 2`
	`0`

Task B2. TOWER

City *X* consists of *N* buildings, ordered in a row from west to east and numbered from 1 to *N*. **Each building has a different height – an integer number, respectively h_1, h_2, …, h_N.** The city government plans to build a tower, which will be in the same row as the buildings (it can be before the first building, between any two of the buildings or after the last building). The tower will broadcast messages to the citizens. **The tower must have height *H*, which should be different from all other buildings' heights**.

Due to some strange engineering ideas, the tower will be able to broadcast signals only to the west (to the beginning of the buildings' row). The signals are also strange – they are rays which travel horizontally (parallel to the ground, which we consider as a straight line) and are emitted out of the whole body of the tower (from the top to the bottom). Therefore, we can imagine that the tower radiates a continuous band of signals with width equal to the tower's height. When a ray hits a building, it stops. **Each building receives the signals using a receiver located on its top.** A building receives a message if at least one ray reaches its receiver.

In other words, a building numbered *i* will receive messages from the tower only when: the building *i* is to the west of the tower; *i* is not higher than the tower; and there is no other building *j* between them ($j > i$), which is higher than building *i.*



Look at the example in the figure above: the buildings, which are able to receive messages, are with numbers 2, 5, 6, and 9.

Write a program **`tower`** to determine the maximum number of buildings, which would receive messages. You will be given the row of buildings in the town (actually, their heights) and the height of the tower. Certainly, you have to consider the optimal placement for the tower.

Input

Two, space separated, positive integers are given on the first row of the standard input: *N* and *H* – the number of buildings and the height of the tower.

N, space separated, positive integers are input from the second row – the heights of the buildings in the town, ordered by the building numbers (from the first to the *N*-th)

Output

On a single row of the standard output print a single integer – the maximal number of buildings, which would receive messages, if the tower were built, assuming optimal placement.

Constraints

$1 \le N \le 1\ 000\ 000$;

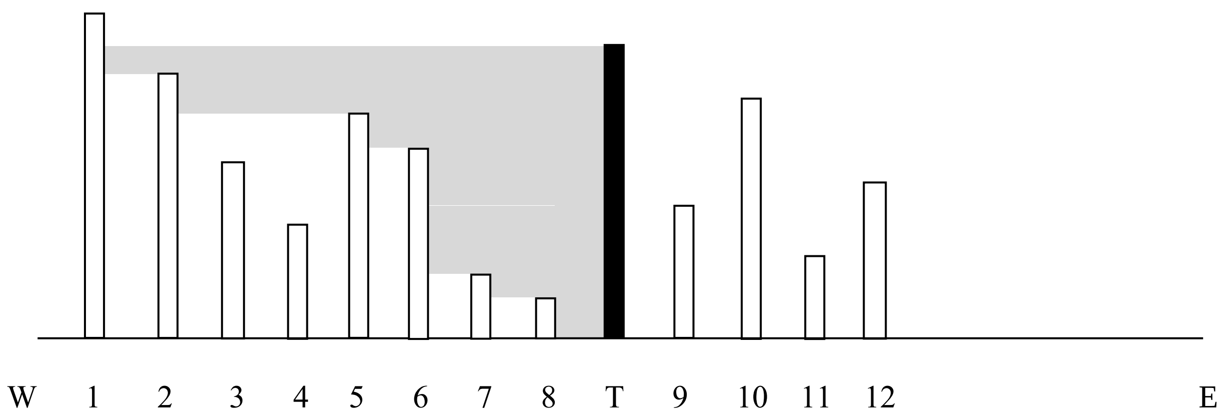
In 30% of the test cases $N \le 1000$;

$1 \le$ height of each building and tower $\le 10^9$

Example

Input	Output
`12 180` `200 170 130 90 150 140 40 30 100 160 50 110`	`5`

Explanation: On the picture below, the optimal location of the tower is given. Messages are received by buildings with numbers 2, 5, 6, 7 and 8.



Task B3. CHEAP TRAVELING

Ivan the Speedy has to pay using his own personal funds the travel to the place of the next programming contest. Moreover, he has only S euro. For that reason, he carefully investigated the schedules of the public transportations and the prices, of course. Let us denote by 1 Speedy's birthplace, by N − the place where the contest will take place, and by 2, 3, …, N – 1 – the other villages he could pass through some of them. In Internet, Ivan found M possibilities for traveling in the form: the bus from village v to village w (as well as from w to v) travels t hours and costs e Euro for each of both directions. There may be more than one bus traveling between v to w, and different buses traveling from v do w may travel for different times and at different ticket prices.

Write a program **`traveling`** to find a trip from village 1 to village N at a cost, which is less than or equal to S Euro. If there exists more than one such traveling, the program must find a traveling in which Speedy will spend minimal time sitting in the busses.

Input

The first line of the **standard input** contains the positive integers S, N and M, $S \leq 2000$, $N \leq 3000$, $M \leq 5000$. Each of the next M lines of the input contains 4 integers − parameters v, w, t and e of one transportation possibility, $1 \leq v \leq N$, $1 \leq w \leq N$, $1 \leq t \leq 100$, $1 \leq e \leq 100$.

Output

The program has to print on a single line of the **standard output** the duration of the found trip. If there no trip of cost less than or equal to S, the program has to print –1.

Examples

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