Problem C. Connectivity

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

This is an interactive problem.

The jury made a random undirected graph of n vertices and m edges with no loops or parallel edges. In each test, the graph is randomly uniformly sampled from all possible graphs with fixed n and m before the testing of your program starts.

Your program has to determine whether the graph is connected, while knowing only n and m at first. The program can perform a "`?` i" query ($1 \le i \le n$) at most $2 \cdot n$ times. In response for such query, the testing system will give either:

- Positive integer j ($1 \le j \le n$) meaning that there is an edge between vertices i and j and that the edge was not previously communicated to the program (including any responses to "`?` j" queries).
- Number -1 meaning that all edges adjacent to the vertex i were already communicated to the program.

Interaction Protocol

The first line of the input contains an integer T: the number of independent test cases for your program to handle. Your program will then have T interactions with the testing system.

An interaction starts with integers n and m, communicated to your program on a separate line: the number of vertices and edges in the secret graph ($2 \le n \le 10^4$, $1 \le m \le \min(\frac{n(n-1)}{2}, 10^4)$).

Afterwards, you can repeatedly print a request on a separate line in the following format: question mark without quotes ("?", ASCII code 63), followed by a space, followed by an integer i ($1 \le i \le n$): the number of the vertex for which you want to know the next adjacent edge.

In response for each such query, the testing system will print either a positive number of another vertex adjacent to i, or -1.

To give the answer, print a line with either "+" (plus sign, ASCII code 43) if the graph is connected, or "-" (minus sign, ASCII code 45) if the graph is not connected.

After printing the answer for the last T-th test case, terminate your solution gracefully.

The sum of n for all test cases in a run does not exceed 10^5.

After printing each line, flush the output buffer, or you will get the outcome `Idleness Limit Exceeded`: this can be done by calling, for example, `fflush (stdout)` in C or C++, `System.out.flush ()` in Java, `flush (output)` in Pascal, or `sys.stdout.flush ()` in Python.

Example

standard input	standard output
2 5 4 2 1 5 -1 3 -1 5 4 -1	 ? 1 ? 4 ? 1 ? 1 ? 4 ? 4 + ? 5 -

Note

Empty lines are added for clarity, they are absent during interaction.

In the example test, the two following graphs are used. The testing system responds to "? i" with the first edge in the list which is adjacent to i and was not yet communicated to the program.

1. $n = 5$, $m = 4$, edges are ordered as follows: 1–2, 1–4, 3–4, 1–5.
2. $n = 5$, $m = 4$, edges are ordered as follows: 1–4, 1–3, 1–2, 3–4.