Ultrafast train

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 512 megabytes

Berland, 2224. In Berland, there are n cities, connected by m railways. Each railway connects two cities and can be used to go in both directions. Also, the railway network includes all cities, so that you can go from any city to any other city by using railways.

Berland railway company (BRC) wants to launch a train that departs with passengers from city 1 and arrives at city n. Since this ride can take a very long time, BRC has decided to speed this train up by installing a time machine.

The time machine works as follows: before departing from any city, it can be turned off or on. If it's turned off, then the ride will take exactly 1 hour. However, if it's turned on, then the ride will take exactly -1 hour, and the train will arrive at the next city earlier than it departed.

BRC wants to plan a route for this train. The route must follow these conditions:

1. The route must start at city 1 and end at city n. All adjacent cities in the route must be connected by railways. It's OK that cities 1 and n could be in the middle of the route.
2. The route must take a nonnegative amount of hours at any point, in order to avoid time-space paradoxes.
3. The route must have at most $3n + 2$ cities, since otherwise the ride is considered complex and unattractive.
4. The route must take the minimal amount of time, by following the conditions above.

Please help BRC to plan the most optimal route. If there are multiple optimal routes, then output any.

Input

The first line of the input contains two integers n and m ($2 \le n \le 10^5, n - 1 \le m \le 10^5$) — the numbers of cities and railways.

The next m lines contain two integers u and v ($1 \le u, v \le n$) — numbers of cities connected by railway.

It is guaranteed that every city is reachable from another city by railways. It's also guaranteed that there are no railways from city to each other and multiple railways between the same cities.

Output

The first line of the output must contain two integers t and k ($t \ge 0, 2 \le k \le 3n + 2$) — minimal time of ride and amount of cities in the route, including first and last cities.

The next line contains $2k - 1$ tokens, depicting the train route. Each 1-st, 3-rd, …, $(2k - 1)$-th element must be an integer, depicting a city. Each 2-nd, 4-th, …, $(2k - 2)$-th element must be `+`, if it's needed to turn off the time machine on this railway, or `-`, if it's needed to turn on the time machine.

Examples

standard input	standard output
5 4 2 1 3 2 3 5 4 2	1 4 1 + 2 + 3 - 5
5 5 1 2 2 3 3 4 4 5 5 2	0 9 1 + 2 + 3 + 4 + 5 - 2 - 3 - 4 - 5

Note

In the first example, the train will arrive in city 2 in one hour, after which it will arrive in city 3 for another hour. After that, the time machine will be turned on and the train will arrive at city 5 in -1 hour. As a result, the ride takes $1 + 1 - 1 = 1$ hour.

In the second example, the train will arrive at city 5 in 4 hours, after which it will go back to city 2, and from there to city 5 in -4 hours. As a result, the ride takes $4 - 4 = 0$ hours.

Scoring

The tests for this problem consist of five groups. Points for each group are awarded only if all tests of the group and all tests of some of the previous groups are passed. Note that passing the samples is not required for some groups.

If the country is **2-colorable**, then each city can be painted in one of two colors, such that each railway connects cities of two different colors.

Group	Points	Additional constraints		Required Groups	Comment
		n	m		
0	0	–	–	–	Samples.
1	17	$n \leq 10$	$m = n - 1$	–	
2	13	–	$m = n - 1$	1	
3	12	–	–	0 – 2	Berland is 2-colorable.
4	26	–	–	–	$u_i = i, v_i = i + 1$ for $1 \leq i \leq n - 1$.
5	32	–	–	0 – 4	