

Navigating Bruteland is very hard. The country consists of n cities connected by m roads. Road number i has color v_i and connects cities a_i and b_i in both directions. That is why, contrary to tree-ish Byteland, there are many ways to travel between selected cities. Fortunately, Eric found a way to determine which one to take – he simply calculates a Phash (Path hash) for each one and selects the one with minimal value. Phash of route (c_1, c_2, \dots, c_k) from c_1 to c_k is equal to $\sum_{i=1}^{k-1} (10^9 + 9)^{(k-i)} \cdot v_{r_i}$ where v_{r_i} is the color of the selected route. Note, that there is no modulo – Eric is smart enough to use Python. Unfortunately that also makes his program, who would guess, quite slow. Help your friend in need and select a route from city 1 to city n with minimal Phash, output its length l (in roads) and a series of values $v_{r_1} \dots v_{r_l}$.

Conditions

- $1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5$
- $1 \leq a_i, b_i \leq n$
- $1 \leq v_i \leq 10^9$
- There can be multiple roads between the same cities.
- There can be roads connecting the city to itself.
- There exists at least one route from 1 to n

Input

```
n m
a1 b1 v1
...
am bm vm
```

Output

The first line of output should contain one integer l - the number of roads in the selected route with a smallest Phash. The second line should contain l values v_{r_1} to v_{r_l} – colours of the selected roads.

Scoring

Subtask	Constraints	Points
1	$n \leq 100, m \leq 200$	12
2	$n \leq 1000, m \leq 5000, v_i \leq 10^6$	13
3	$n \leq 1000, m \leq 5000$	15
4	no additional constraints	60

Example

Input	Output
4 6 1 2 1 1 3 2 3 4 3 2 3 1 2 4 4 3 1 1	2 1 3
2 1 1 2 1	1 1
4 4 1 2 1 1 3 1 2 4 3 3 4 2	2 1 2