

E. Pumping Stations

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Mad scientist Mike has applied for a job. His task is to manage a system of water pumping stations.

The system consists of n pumping stations, which are numbered by integers from 1 to n . Some pairs of stations are connected by bidirectional pipes through which water can flow in either direction (but only in one at a time). For each pipe you know its bandwidth — the maximum number of liters of water that can flow through it in one hour. Each pumping station can pump incoming water from some stations to other stations through the pipes, provided that in one hour the total influx of water to the station is equal to the total outflux of water from the station.

It is Mike's responsibility to pump water between stations. From station a to station b through the pipes (possibly through other stations) within one hour one can transmit a certain number of liters of water according to the rules described above. During this time, water from other stations can not flow into station a , and can not flow out of the station b . However, any amount of water can flow out of station a or in station b . If a total of x litres of water flows out of the station a in an hour, then Mike gets x bollars more to his salary.

To get paid, Mike needs to work for $n - 1$ days, according to the contract. On the first day he selects two stations v_1 and v_2 , and within one hour he pumps a certain amount of water from v_1 to v_2 . Next, on the i -th day Mike chooses a station v_{i+1} that has been never selected before, and pumps a certain amount of water out of the station v_i to station v_{i+1} for one hour. The quantity of water he pumps on the i -th day **does not depend** on the amount of water pumped on the $(i - 1)$ -th day.

Mike needs to earn as much bollars as he can for his projects. Help Mike find such a permutation of station numbers v_1, v_2, \dots, v_n so Mike will be able to earn the highest possible salary.

Input

The first line of the input contains two space-separated integers n and m ($2 \leq n \leq 200$, $1 \leq m \leq 1000$) — the number of stations and pipes in the system, accordingly. The i -th of the next m lines contains three space-separated integers a_i, b_i and c_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$, $1 \leq c_i \leq 100$) — the numbers of stations connected by the i -th pipe and the pipe's bandwidth, accordingly. It is guaranteed that any two stations are connected by at most one pipe and that there is a pipe path between any two stations.

Output

On the first line print a single integer — the maximum salary Mike can earn.

On the second line print a space-separated permutation of n numbers from 1 to n — the numbers of stations in the sequence v_1, v_2, \dots, v_n . If there are multiple answers, print any of them.

Examples

input

```
6 11
1 2 10
1 6 8
2 3 4
2 5 2
2 6 3
3 4 5
3 5 4
3 6 2
4 5 7
4 6 2
5 6 3
```

output

77
6 2 1 5 3 4