

# Travel Dream

Input file:            **standard input**  
Output file:         **standard output**  
Time limit:          3 seconds  
Memory limit:       256 megabytes

Andy is an ordinary student at Nanjing University. One day, he found himself transmitted to a wonderland when he was dreaming. The wonderland consisted of several spots, with some bidirectional roads of various travel times connecting some pairs of them.

Attracted by the fascinating scenery, Andy wanted to visit some spots in the wonderland. He would like to select exactly  $k$  distinct spots and visit them one by one. After visiting the last spot, he must turn back to the first spot, because that was where his dream started. There must be a road between any two adjacent spots he selected, including the last one and the first one. Moreover, he wanted to maximize the total time spent in moving between the spots, so that he could better enjoy the beauties.

For example, in the first sample data, you may choose to start from spot 1, visit spots 3, 5, 2, and return to spot 1. The total time spent was 21 minutes, which turns out to be optimal.

However, finding such a traveling plan was too hard for Andy, so he wanted your help. Could you help him to find such an optimal plan?

## Input

The first line of input consists of three integers  $n, m, k$  ( $2 \leq n \leq 300, 1 \leq m \leq 300, 3 \leq k \leq 10$ ), denoting the number of spots and the number of roads in the wonderland, and the number of spots Andy would like to visit, respectively. The spots are numbered 1 through  $n$ .

The remaining  $m$  lines specify the roads between the spots. Each of these lines contains three integers  $u, v, t$  ( $1 \leq u, v \leq n, u \neq v, 1 \leq t \leq 10^8$ ), representing a bidirectional road between spots  $u$  and  $v$ , and it took  $t$  minutes to travel along the road in either direction. It is guaranteed that there was at most one road between any pair of spots in the wonderland.

## Output

Print the maximum total travel time in minutes as an integer in one line. If it is impossible to find any valid travel plan, output **impossible** instead.

## Examples

standard input	standard output
5 7 4 1 2 2 1 3 3 2 3 4 4 3 1 5 3 7 4 5 6 2 5 9	21
5 4 5 1 2 1 2 3 6 3 1 5 4 5 2	impossible