Problem I. Knapsack Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Given an undirected graph with n vertices and m edges, each edge holds an item with weight w_i. Xiao S has a compulsion: every time he traverses an edge, he must put the item on that edge into his backpack. The backpack has a capacity V. Initially, he starts with a new backpack. If the remaining capacity of the current backpack is insufficient to carry w_i, he switches to a new backpack (discarding the old one). For each starting vertex from 1 to n, Xiao S chooses a path to reach a designated vertex T. Your task is to determine the minimum number of backpacks needed for each starting vertex to reach T. If it is impossible to reach T, output -1.

Input

The first line contains four integers n, m, V, and T ($1 \le n, m, V \le 10^5, 1 \le T \le n$).

The following m lines contain three integers each; the i-th line contains three integers x_i, y_i, and w_i, representing an edge between x_i and y_i with an item of weight w_i. It is guaranteed that $1 \le x_i, y_i \le n$ and $1 \le w_i \le V$.

Output

Output a single line with n integers; the i-th integer is the minimum number of backpacks needed when starting from vertex i. If T is unreachable, output -1.

Example

standard input	standard output
8 10 7 4 1 2 4 2 3 4 3 4 4 1 5 2 5 6 5 6 7 3 7 4 4 2 6 3 3 6 1 2 5 3	2 2 1 1 2 1 1 -1