Programming Assignment 3

Introduction

The advent of smartphones has made GPS navigation commonplace. Most routing and ride-hailing applications already employ their own "best" route algorithm when traveling from one place to another.
The simplest way to select the "best" route is by selecting the route with the least amount of distance to travel. The shorter the route the better. However, if most applications display or recommend the shortest route from one place to another, it is likely that the said route will incur more traffic as more people are selecting that route through their apps. This will defeat the purpose of selecting it as the "best" route in the first place.

A naive entrepreneur has just recently discovered this exact problem and wants to develop a prototype of a routing application that will return what s/he calls "the route less traveled". That is, the minimal-length route that does not use any of the road taken by the shortest route connecting two places.

Problem Description

With no programming background, the naive entrepreneur has contracted you to develop a prototype of his/her desired routing application. The routing application will accept two input coordinates: (1) a starting point s; and (2) a destination point t. For prototype testing, you are given the map data of a test region which has V coordinate nodes corresponding to different landmarks, pick-up points, and drop-off points. Each coordinate node in the map has an ID number from 0 to $V-1$. Two distinct coordinate nodes may be connected directly by a one-way road link. You can easily determine that there are E one-way road links connecting the various nodes in the said map you are provided.

You are also given the following specification for special cases that the routing algorithm may encounter: (1) if there are multiple shortest routes, you cannot use any of the road links taken by any of the shortest routes; (2) you are also to specifically flag if there are no possible route that will not use any of the road links used by the shortest route(s); and (3) if there are multiple minimal-length routes that do not pass through the shortest route, you only need to show one of them.

Input Description

The first line of the input contains two natural numbers $2 \leq V \leq 500$ and $1 \leq E \leq 10000$. The second line contains two natural numbers: the starting node, s, and the destination node, t. The last E lines each contain three natural numbers u, v ($u \neq v$), and $1 \leq c \leq 1000$ which indicates the existence of a one-way road link from node u to node v with length c. Multiple entries in a line are separated by spaces.

Output Description

The first line of the output contains the length of the desired route. The second line contains the desired minimal-length route itself, denoted by the node ID of the coordinates in the map, separated by spaces.

Grading Milestones

20%	able to find the standard shortest route(s) from the input
60%	able to process the sample test cases
80%	able to process all provided test cases
100%	able to handle additional test cases not shown
100%+bonus	able to toggle routing behavior

Sample Input/Output 1

```
7 9
0 6
0 1 1
0 2 1
0 3 2
0 4 3
1 5 2
2 6 4
3 6 2
4 6 4
5 6 1
```

```
5
0 2 6
```

Sample Input/Output 2

```
4 6
0 2
0 1 1
1 2 1
1 3 1
3 2 1
2 0 3
3 0 2
```

```
-1
-1
```

Bonus

It has been discovered that the prototype algorithm tends to find no valid routes for nearby places. Create a separate file that modifies the routing algorithm to allow the desired output route to share at most one (1) one-way road link with the shortest possible route. The restriction being that this shared one-way road link must have length less than 20% of the resulting new valid route.

Notes

1. **DO NOT** use tabs to separate output.
2. If you have a question involving source code, please do post the code snippet and/or error messages. Always resolve compile-time errors in the order they appear in your code. If you think that the code snippet might be too revealing of your work, post it as a private question first and I'll decide if said code may be made public to your classmates.
3. Make sure that you are using a g++-8 compiler which is either from GCC 8.1 or GCC 8.2 so that you will not have issues during checking. If you want to use a new C++ specification kindly comment what compile command you are using as a comment at the beginning of your source code.
4. You may use any library in the C++ STL.
5. If you have a question just post it at Piazza for better traction.
6. Several test input/output will be provided three full days before the soft deadline; however, sample input/output for the bonus will not be provided.

Submission

1. Soft deadline is set to 9:00 PM of 19 April 2019.
2. Hard deadline is set to 9:00 PM of 27 April 2019.