CS 201 Data Structures and Algorithms
Autumn 2021

Main Quiz#2 Part-2	
Objective	**To implement Bellman Ford algorithm**
Total marks	**5**
Due date (without penalty)	**6th November (Saturday) 11:59 pm**
Cut-off date (with penalty - 5%)	**10th November (Wednesday) 11:59 pm**
Penalty for violating naming convention(s)	**5%**

Input
Your program should accept an input file as a command-line argument. A typical execution of your program will be *./a.out sample.graph*. The input file represents a weighted directed graph. The first line of the file contains two numbers - first being the *number of vertices n* and the second being the *number of edges m*. The vertices are numbered from *0 to n-1*. Every other line is of the form u v w, which represents a directed edge from vertex u to vertex v where the weight of the edge is w. Please note that there may be vertices in the graph not part of any edges.

Task
Implement Bellman-Ford algorithm to find shortest path distances from the *source vertex 0 to all vertices*. The output file should be named as ***'sd.txt'***. The output file must contain exactly *n lines* and each line is of the form *<vertex> <dist>*, where <vertex> is the label of the vertex (this value is 0 in the first line, 1 in the second line and so on) and <dist> is the distance from the vertex 0 to <vertex>. The value <dist> can be of three types:

(i) An **integer** - if there is a path from vertex 0 to <vertex> and the shortest distance is well defined;
(ii) "***+inf***" if there is no path from the vertex 0 to other <vertex>;
(iii) "***-inf***" if there is a path from the vertex 0 to <vertex> but the shortest distance is not well defined *(because there is a path from 0 to <vertex> which contains a negative weighted cycle)*

Submission

- The program you submit should output '***sd.txt'*** when run.
- The main file of your program should be named as <roll no>.<extension>, where roll no. specifies your roll no. and the extension depends on the language you choose (Usage of C is mandatory for this assignment). Ex: 200010001.c.
- Test well before submission. You may use the attached sample input file for testing. The corresponding output file is also attached. We have some hidden inputs with us to test your program. The mark you obtain is purely based on whether your program correctly gives outputs for the hidden inputs.

- If your program has only a single source file, please submit the file as it is. If your program has multiple source files, please submit your code as a zip file where the name of the zip file should be your roll number. It is important that you follow the input/output conventions exactly (including the naming scheme) as we may be doing an automated evaluation. ***There will be a penalty of 5% (on the mark you deserve otherwise) if you do not follow the naming conventions exactly.***
- Follow some coding style uniformly. Provide proper comments in your code.
- Submit only through moodle. Submit well in advance. Any hiccups in the moodle/internet at the last minute is never acceptable as an excuse for late submission. Submissions through email or any other means will be ignored.
- Acknowledge the people (other than the instructor and TA) who helped you to solve this assignment. The details of the help you received and the names of the people who helped you (including internet sources, if applicable) should come in the beginning of the main file as a comment. Copying others' programs is a serious offence and a deserving penalty will be imposed if found.

Evaluation

- To consider for the evaluation without penalty, you have to submit your program by the due date. If you submit after the ***due date*** but on or before the ***cut-off date***, ***there will be a penalty of 5% on the marks you deserve otherwise***.
- If you do not submit by the ***cut-off date,*** your program will not be considered for the first evaluation.