Write a program in C/C++(only) to find the Critical Path in an activity graph.

(explanation of critical path is given in section 3.1, p.87)

The program should also be able to find "Slack Time" for every path.

Given, a duration in days, the program should be able to find the Feasibility of the project completion in that duration.

Due : Thursday, October 12, Midnight

Your program should take following as the inputs :

- Number of Activities(Vertices)
- Number of Edges
- Weights(duration in days) for each edge. An edge can be represented as triplet(activity***1***, activity***2***, duration).
- Duration(In days).

Your program should be able to generate following outputs

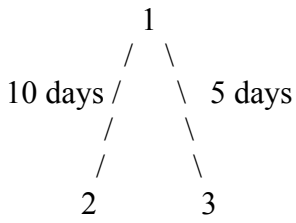
- Display all paths along with their slack times
- Mark critical paths distinctly
- Minimum time required to complete the project
- Feasibility of project completion (YES/NO)

Input Formats:

- First line specifies the number of activities(vertices)
- Second line specifies number of edges
- succeeding lines contain information of all the edges in the form(activity***i***,activity***j***,weight)
- last line specifies, duration in days

See example below

e.g for the graph and duration 20 days, activities 1,2 and 3



Sample Input would be :

3
2
1,2,10
1,3,5
20

Output formats:

1. Lines in the beginning contain all the paths, a critical path should be marked by CRITICAL at the end and non critical path should be marked by SLACK(t), for that path. Now, there can be multiple critical paths.
2. Second line specifies minimum time required to finish the project(in days)
3. Last line contains feasibility of the project completion in given duration

Sample output for the above graph would be:

1,2,CRITICAL
1,3,SLACK(5)
10
YES

Programs must be done in C/C++. No other language should be used. Also, programs should get compiled and linked on gl machines. Programs should not have any GUI, but should take input from stdin and should print output on stdout.

(Note : No fancy input / output is expected, stick to the given input and output formats, as your programs will be tested automatically in most cases. Do not forget to submit a readme.txt including a statement of help, briefing the logic of the program and also specify if your program behaves differently than what is required)

Instructions to submit the code will be announced on the course webpage, so keep checking it periodically.

Click here to see a sample program that accepts inputs in specified format