

**CS69003: Computing Systems Lab 1**  
**Autumn 2007**

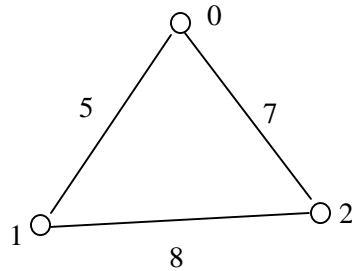
**Assignment 1**

**Implementing some basic graph algorithms**

**Due: August 1**

In this warmup assignment, you will implement some basic graph functions. In particular, you will read in an undirected weighted graph from a file, check and print if it has any cycles or not, and given any two nodes of the graph, print the shortest path between them

The input graph is given in a text file in the following format. The first line of the file contains the number of nodes  $n$ . The nodes are numbered from 0 to  $(n - 1)$ . Thereafter, each line contains one edge, given as a sequence of three integers separated by one or more spaces, the two node numbers and the weight of the edge. For example, consider the graph below.



The file corresponding to the graph will look like

```
3
0 1 5
0 2 7
1 2 8
```

In your program, you will first implement a type called GRAPH that can store a graph of arbitrary number of nodes and edges. Then write the following functions with the given prototypes:

1. *int ReadGraph( GRAPH \*G, char \*inp\_file)* – reads in a graph from the file named *inp\_file* in the graph *G*. The file *inp\_file* should have a graph in the above format. Returns 0 if graph is read successfully, -1 otherwise (for ex., file not present).
2. *int IsCyclePresent(Graph G)* – returns 1 if the graph contains a cycle, 0 if the graph does not contain a cycle, and -1 if there is an error.
3. *int PrintShortestPath(Graph G, int x, int y)* – prints the shortest path from node *x* to node *y* in graph *G*. Returns 0 on success, -1 if there is any error. Note that you

need to actually print the sequence of nodes in the path starting with  $x$  and ending with  $y$ , not just the length of the path.

Your program will take one command line argument, the input file name. Your *main()* function will first read in the input file and check and print if it contains a cycle (the printing must be done from within *main()*, not from *IsCyclePresent()*). It will then enter a loop in which it will prompt for two node numbers from the user. It will then print the shortest path from the first node to the second. The program terminates if any of the two node numbers entered is negative.

Design appropriate data structures to define GRAPH and implement the above functions. You should submit the following file.

1. A .c file containing your program. Name the file <your roll no.>\_graph.c (for ex., 07CS1004\_graph.c).

***It is very important that you follow the above file naming convention and function prototypes EXACTLY as the evaluation may be done by a program that will assume these. Any error arising out of deviations from above will incur severe penalty in marks.***