A transaction in a database management system is a sequence of database operations executed from a database application or by a user interacting with the system. These operations must be executed all entirely or not executed at all. Several transactions can be executed concurrently. Transaction may requests locks to database objects. A database object could be an entire database record, a single database field, or an entire database table. A deadlock occurs when two or more transactions are waiting for locks held by the other to be released. To detect a deadlock, a wait-for graph (WFG) is built. In a WFG, nodes are transactions, and an edge is created from transaction Ti to Tj when Ti is waiting to lock a database object locked by Tj. Deadlock exists if there is a cycle in the WFG.

The system generates a series of lock (e.g., LOCK T1,A), unlock (e.g., UNLOCK T1,A), and CHECK SYSTEM commands. Example of a set of commands are:

LOCK T17, A; LOCK T17, B; UNLOCK T17, A; LOCK T18, A; LOCK T20, C; LOCK T21, D; LOCK T21, E; LOCK T20, E; LOCK T18, C; LOCK T21, A; LOCK T22, E; LOCK T22, B; CHECK SYSTEM

The above lock and unlock commands generate the following WFG:

Write a program that reads a simplified version of these commands from a file, one on each input line (L 17 A; L 17 B; U 17 A; L 18 A; etc.), build the graph and detect if there is a cycle by trying to perform a topological sort. A cycle is present in the WFG if the sort cannot be performed. Display one of these two messages:

1) There is no cycle in the WFG
2) WARNING: There is a cycle in the WFG.

You could work with another teammate on this project. Submit the source code of the program and two test cases using the department submission website. One test case must have a cycle and the other case without cycles. You must have at least 5 records on each test case. Please describe the test cases on a separate document.