2006/2007 SOUTHERN CALIFORNIA REGIONAL
ACM INTERNATIONAL COLLEGIATE PROGRAMMING CONTEST

Problem 6
CMDB

Swamp County College is embarking on an effort to improve the management of the College's data center. The college is adopting the ITIL (Information Technology Information Library) model for IT service management. Originally developed by the British Government, ITIL covers a wide spectrum of IT service, application, and infrastructure management best practices. ITIL is now being adopted widely in both the public and private sectors worldwide.

The college IT department has determined that it needs to improve its understanding of the infrastructure within the data center. In accordance with the ITIL model, the IT managers want to build a Configuration Management Data Base (a CMDB) that contains information about the equipment in the data center, the connections between that equipment, and the databases and software applications that run on the equipment. The objective is to be able to identify the potential impact of downtime or maintenance of one infrastructure component on the databases and applications supported by the infrastructure.

Building a full CMDB is a major undertaking. In order to get some value out of the process quickly, and gain executive support and funding for the full project, the college is asking your team to develop a proof-of-concept database and tool that will demonstrate the basic functions of a CMDB and help them when planning upgrades to their critical infrastructure.

This proof-of-concept database will handle the following infrastructure components and the relationships between them:

- Storage Arrays
- Storage (Fibre Channel) Switches
- Network (IP) Switches
- Servers
- Databases
- Applications

Each of these infrastructure components is referred to by ITIL as a configuration item (CI). The database will contain a type definition of each CI and a list of relationships between them. Each CI will be defined as one of the above types (storage array, storage switch, network switch, server, database, or application). CI names are strings of between 1 and 31 upper case letters and digits, with the first character being a letter. No two CIs will have the same name.

Relationships will be defined as connections or usage relationships between CIs; for example, a storage array connects to a storage switch, a server connects to a network switch, a server may connect to a storage switch, an application runs on a server, an application may use a database, an application may use another application, and a server may use a storage array. Databases and applications that depend on each other may reside on different servers, communicating with each other over the IP network.

Storage switches provide connections between a server and a storage array. Every path between a server and its external storage will pass through only one storage switch. However, connections to storage arrays may be redundant—that is, a given server may have independent connections to multiple switches, as may a storage array. This means that there may be multiple paths between a server and a storage array. These redundant paths are used to provide additional bandwidth and redundancy, so that if one storage switch fails or is taken down for maintenance, access to storage is not interrupted.

Similarly, servers may have connections to multiple network switches, providing multiple paths for network traffic among databases and applications. Unlike storage, there may be multiple IP switches in a path between server network ports.

Problem 6
CMDB (continued)

Your team is to write a program that will read CMDB entries, then use the CMDB entries to answer queries about the impact that a single down CI would have on database and application CIs.

Input to your program will be a series of CI definitions, followed by a series of CI relationships, and ending with a series of queries. Each definition, relationship, and query appears on a line by itself, with the fields separated from each other by one or more spaces. No input line will exceed 80 columns.

Each CI will be defined on a line containing the string "CI", the CI name, and the type:

- storage-array
- storage-switch
- network-switch
- server
- database
- application

Each relationship will be defined on a line containing the string "RE", a CI, the type of relationship (as listed below), and another CI. The relationship definitions are as follows:

- $\langle storage\text{-}array \rangle$ connects-to $\langle storage\text{-}switch \rangle$
- $\langle server \rangle$ connects-to $\langle storage\text{-}switch \rangle$
- $\langle server \rangle$ connects-to $\langle network\text{-}switch \rangle$
- $\langle network\text{-}switch \rangle$ connects-to $\langle network\text{-}switch \rangle$
- $\langle server \rangle$ uses $\langle storage\text{-}array \rangle$
- $\langle database \rangle$ runs-on $\langle server \rangle$
- $\langle application \rangle$ runs-on $\langle server \rangle$
- $\langle application \rangle$ uses $\langle database \rangle$
- $\langle application \rangle$ uses $\langle application \rangle$

The CI relationships will be fully defined—there will be no missing paths between components. Each application and database runs on only one server. Any application may use the services of several other applications and databases.

Queries to the CMDB will begin with "??" followed by a CI that has failed or is scheduled for maintenance. Your program is to print the CI being queried, followed by a list of database and application CIs that would be affected. The CI being queried is to be printed on a line with the string "Query:" starting in the first column, followed by a single space and the name of the CI. The affected database CIs are to be printed next, each on a separate line that begins with "Database:" followed by a single space and the affected CI name. Applications are to be listed last, using the same format—each on a separate line that begins with "Application:" followed by a single space and the affected CI name. The database and application CIs should be listed in ASCII lexical order. The lines containing affected CI names are to begin in the fourth column. If a CI is listed because it is in a degraded state, print an asterisk in the third column.

A CMDB storage switch query is to report that the databases and applications running on a given server are running in a degraded state if the server can reach all its storage on one or more other paths. Should such a server lose all connectivity to a storage array it uses, all databases and applications running on that server will be down. If a server should lose a network connection that affects needed connectivity for applications that run on the server, those applications will be degraded or down (based on the availability of other network paths).

Your proof-of-concept implementation is to have the capacity to handle 250 CI definitions and 1,000 CI relationships.

Problem 6
CMDB (still continued)

Sample Input

```
CI AA storage-array
CI S1 server
CI S2 server
CI S3 server
CI FCSW storage-switch
CI IPSW network-switch
CI ODB database
CI PDB database
CI FINANCE application
CI CRM application
RE CRM uses ODB
RE FINANCE uses PDB
RE S1 uses AA
RE S2 uses AA
RE S1 connects-to FCSW
RE S2 connects-to FCSW
RE AA connects-to FCSW
RE ODB runs-on S2
RE PDB runs-on S1
RE S1 connects-to IPSW
RE S2 connects-to IPSW
RE S3 connects-to IPSW
RE FINANCE runs-on S1
RE CRM runs-on S2
?? PDB
?? S2
```

Output for the Sample Input

```
Query: PDB
   Application: FINANCE
Query: S2
   Database: ODB
   Application: CRM
```