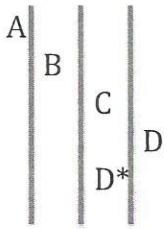
Maintenance Project Documentation

The maintenance project requires us to look at transactions and modules in a system and their relationships with each other. The transactions are the roots and from them all the other modules are connected. Traversing to any module begins with the transactions. Modules are able to call other modules. For example, BC would show that module B has module C as a child module, or in other words B can call C. Each module is able to call any number of other modules. Modules can be called by different modules. A module cannot call any module that has already called it, or was called before it. This way there are no infinite loops in any of the relationships in the system.

The program has certain task that it must be able to complete. The program console should **not** interface with the user. The user is able to input a defective module, or a transaction. Each input will yield a different output. Also program should be able to show each of the unique modules in the system excluding the transactions. For example, if the system is AB; AC; AD; BX; CY; DZ the program would print B, C, D, X, Y and Y. Another task the program must be able to do, is show the path to a given defective node. For example, if the system is AB; BC. Next the user inputs that the defective node is C. Then the program should print out A B C. The final requirement for the document is for it to be able to show the explosion of a given transaction. When the user inputs a given transaction, the program should show all of the different paths to its connecting modules. For example, if the system is AB; BC; BD; CD. The given transaction by the user is A. Then the explosion output should be:



Specification Details:
-read from text file
-one character per module
-one character per transaction
-one or more transactions
-transactions do not appear on right side
-tell how many transactions (first)
-explosion then defect in file input
-modules will be provided by Keiller
-students may choose to code in C++ or Java
-for defective modules, print path to transactions
-requirements, assumptions, pseudo code, test cases, and source code should be documented
-semicolon separation

Format of Text File:

AB;
BC;
BD;
A; //module expected to be exploded
D; //defective module

Assignment is due 2 weeks before the semester's end.

I PETER A. KEILLER agree to the requirements detailed on this paper
(Please Print)

for the Maintenance Project.

X