Design #5
Author: Alves Silva, Otavio Augusto
Date: 06/03/2015
CS 163 - Program #5

1) How well did the data structure selected perform for the assigned application?

The table(Graph) selected for this assignment was able to achieve the objective that was assigned. The graph is totally capable to work with a previous provided amount of memory that the user will provide. The table was capable to organize hierarchically the data without has to use a lot of operators to executes it. Therefore, the data structure proved be a good way to organize hierarchically the data and simulates a graph of tasks.

2) Would a different data structure work better? If so, which one and why?

Probably, a different data structure would not work better. The assignment for this program has some specifications that would be hard to implement in a different data structure. Imagine that we had to create a pointer to each node in the graph using a tree or a LLL. With graph, we have to manage just two kinds of structures that are pointers of them. Therefore, a different kind of data structure would be a bad choice to implement in the program.

3) What was efficient about your design and use of the data structure?

The graph was capable to achieve all requirements and functions to be considered efficient. The "insert edge" function was able to insert a link between the tasks any time during the execution of the program. The "display adjacent" function was able to display all the adjacent tasks of a specific task using a provided task name. The data structure proved a good choice to organize data with multiples links and relations.

4) What was not efficient?

The "insert edge" function inserted a relation link between the vertices of the list, but it just happens when the user executes the command that does that and the vertices, which are going to be by this function, must be created before it can be executed. Therefore, it could be annoying for the user because we have to add every task and after that have to insert the edges of the tasks.

5) What would you do differently if you had more time to solve the problem?

I would find a way to solve the insertion of edges in a more practice form. Maybe, in the moment that the user inserts a new task, I would ask to him if that task has a requisite or if that task is a pre-requisite of any previous task. Therefore, the program would do the insert edges automatically without any prejudice for the system.