The following exercise should be implemented in python, using built-in packages only.

Your solution should be tested.

Definitions:

Task:

A task status can be pending/in progress/done.

each task has a priority 0-9.

The task can point to an unlimited amount of other (dependent) task. Assume no cycle exists.

Task Executor:

It gets a list of tasks with task dependencies. Takes ownership of the task, executes them in the correct order, and proceed to the next task.

1) Implement Task class.

2) Implement Task Executor allowing the functionality of different execution order of tasks:

In all the cases, a task can't be executed if it has dependent tasks not in "done" status.

a) First option should execute all tasks with no dependencies and then execute the tasks with dependencies.

b) Second option should execute all the tasks based on priority first.

3) Add a functionality of finding if there is a cycle in your tasks. In case a cycle is found, print an indicative message and abort before starting execution.