Dependency Schedule

step 0: prepare everyhing
- numberNodes <-- get from input (1)
- totalJobTime <-- the sum of all jobs, obtain from input (2)
- ProcNeed <-- get from input (3)
- if (procNeeds > numberNodes)
  procNeeds <-- numberNodes
- dynamically allocate accordingly and initallizing all arrays.
- print all three inputs to the output file
- ProcUsed <-- 0
- Time <-- 0

step 1: 1.1: orphenNode <-- find an unmarked node on the dependancy graph that does not have any parent
(ie., check parentCount[i] == 0 and jobMarked[i] == 0)
1.2: mark orphenNode (i.e., update jobMarked)
1.3: put orphenNode onto OPEN list
1.4: repeat step 1.1 to 1.3 until all jobs are checked.

step 2: availProc <-- find an available processor within the used processors
(looking into processjob[i] <= 0, i from 0 to ProcUsed;
if there is un-occupied processor within ProcNeed,
returns the processor ID, otherwise
returns -1 // all Procs are occupied

if availproc >= 0
newJob <-- remove from OPEN place newJob on the processJob[availProc],
place newJob's time on processTime[availProc]
update scheduleTable on the availProc row,
(with respect to TIME status and job's time requiements)
procUsed++

step 3: repeat 2 until OPEN is empty or ProcUsed >= ProcNeed

step 4: if OPEN list is empty
and D.G is not empty
and all processors finished all the jobs
report error and exit (there is cycle in the graph)

step 5: print to the *console* the scheduling table,
TIME, ProcUsed, and all 1-D arrays with proper heading. // for debugging

step 6: Time++

step 7: Decrease all processTime[i] by 1

step 8: job <-- find a job that is done, ie., processTIME [i] == 0 ;
delete the job from the processJob[i] (update processJob[i])
delete the job from the graph (update jobDone[job])
delete all it's outgoing arcs
(decrease by 1, the paraentCount[job] of its dependents)
jobDone[job] <-- 1
procNeed--

step 9: repeat 8 until no more finished job

step 10: print to the *console* the scheduling table,
TIME, ProcUsed, and all 1-D arrays with proper heading. // for debugging

step 11: repeat step 1 to step 10 until the graph is empty
(looking into the 1-D array of jobs'status)

step 12: print final scheduleTable to the output file.