

Problem 1:

In a particular game, n participants decide to arrange themselves in a circle in positions numbered from 0 to $n-1$. The game goes as follows: Starting from a given participant, a judge starts counting participants and removes the m -th person from the circle. Then, the judge starts counting again going around the circle until only one participant, the winner, is left. Write a C++ program that takes n and m as integers and determines which position corresponds to the only player remaining in the game. HINT: Use a circular queue where after all the participants are added, the head is connected to the tail.

Problem 2:

You work for a company that creates a process where n jobs have to be sequenced. Let's number these jobs from 0 to $n-1$. There are dependencies between these jobs, that is some jobs must be completed before other jobs can be started. The parameters to this problem are: n : the number of jobs, then for every job: a tuple $t(i,j)$ that specifies that job i depends on job j . In other words, job(j) must be completed before job(i) can be started. There may be more than one tuple or dependency relationship for any given job, that is a job may have more than one dependency. Your task is to write a program to schedule the execution of these jobs. In case of a tie, choose any jobs in the tie arbitrarily. HINT: For each job i build a queue of jobs that depend on it, that is, jobs that must be scheduled after job i . Then, for each job, count the number of dependencies that job has. Then, create a queue the *ready* queue that contains all jobs that have no dependencies. Select one job from this queue, this is the current or active job, and remove 1 dependency from every job that depends on this job. You can also delete this job. When the dependency count for any given job reaches 0, add that job to the ready queue. Proceed until all jobs are executed. A tuple of the form (i,i) indicates that the job has no dependencies.