**problem**

A university computer science department has a teaching assistant (TA) who helps undergraduate students with their programming assignments during regular office hours. The TA's office is rather small and has room for only one desk with a chair and computer. There are three chairs in the hallway outside the office where students can sit and wait if the TA is currently helping another student. When there are no students who need help during office hours, the TA sits at the desk and takes a nap. If a student arrives during office hours and finds the TA sleeping, the student must awaken the TA to ask for help. If a student arrives and finds the TA currently helping another student, the student sits on one of the chairs in the hallway and waits. If no chairs are available, the student will come back at a later time.

**Simple process flows**

1. make student threads and ta thread (initialize)
2. each thread is running own thread function (student is programming, ta is sleeping)
3. student will wake the ta up after randomly time, then ta will help the arrived student (change student's semaphore to 1 and wait ta's semaphore). but if chairs are full, students go to their programming back and return again after randomly time
4. if ta's help is finished, ta check the remaining students. if there is(are) student(s) ta is keeping the current state (help continuously) and if not, ta go to sleeping back (change ta's semaphore to 1 and wait student's semaphore)
5. repeat 3-4 steps

**Functions**

1. stu\_programming: students is waiting in random time (they are programming in this time), and notice to ta for help. if ta is helping a student already, other students are waiting chairs in hallway, but if chairs is full, students go to their programming task back
2. ta\_teaching: ta is sleeping until notice is arrived from at least one student. if any student wake the ta up, ta will help the student. if helping is finished and chairs are empty, then ta go to sleeping back
3. rand\_sleep: make sleeping time (it means programming task or helping tesk) randomly

**solution**

Using Pthreads, n students are created. Each student, as well as the TA, run as a separate thread. Student threads alternate between programming for a period of time and seeking help from the TA. If the TA is available, they obtain help. Otherwise, they either sit in a chair in the hallway, or if no chairs are available, resume programming and seek help at a later time. If a student arrives and notices that the TA is sleeping, the student notifies the TA using a semaphore. When the TA finishes helping a student, the TA checks to see if there are students waiting for help in the hallway. If so, the TA helps each of these students in turn. If no students are present, the TA returns to napping.

The TA and each student thread print their state and threadID (student number).

The program can take 0 or 1 command line parameters. The number of students present can be specified as a command line parameter. If this parameter is not included, the number of students defaults to 5.