```
/* ----------------------------------------------------------- */
/* NAME : Bryan Wandrych                        User ID: bdwandry */
/* DUE DATE : 12/4/2021                                        */
/* PROGRAM ASSIGNMENT 4                                        */
/* FILE NAME : thread-main.cpp                                 */
/* PROGRAM PURPOSE : The purpose of this program is to solve   */
/* the Party Problem by implementing 2 different types of      */
/* threads, students and landlord. Students                    */
/* will join/leave a room freely. However, if the landlord     */
/* appears at a random time and the room is over capacity, they*/
/* will remove everyone from that room. This file will act     */
/* as the starting point for the rest of the program, it will  */
/* spawn the threads accordingly.                              */
/* ----------------------------------------------------------- */
#include <iostream>
#include "thread.h"
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
using namespace std;

/* ----------------------------------------------------------- */
/* FUNCTION: spawnThreads                                      */
/*    The pourpose of this function is to spawn the threads    */
/*    for the Students and Landlord                                            */
/* PARAMETER USAGE :                                           */
/*    int m: number of times the landlord checks the room.     */
/*    int n: maximum number of students allowed in the room.   */
/*    int k: total number of students                          */
/* FUNCTION CALLED :                                           */
/*    Main                                                     */
/* ----------------------------------------------------------- */
int spawnThreads(int m, int n, int k) {
        //Initializer for the threads
        LandLordThread *landlordThread = new LandLordThread(m, n);
        StudentsThread *studentsThread[k];

        //Spawns the Landlord Thread
        landlordThread->Begin();

        //Spawns Student Threads
        for (int i = 0; i < k; i++) {
                StudentsThread::totalProcesses++;
                studentsThread[i] = new StudentsThread(i);
                studentsThread[i]->Begin();
        }


        //Waits for all the students threads to exit gracefully
        for (int i = 0; i < k; i++) {
                studentsThread[i]->Join();
        }

        //Waits for the landlord thread to exit gracefully
        landlordThread->Join();

        return 1;
}

/* ----------------------------------------------------------- */
/* FUNCTION: main                                              */
/*    This function is the main function to this program       */
/*    It will start out by reading inputs from arguments and   */
/*    spawn later process off of those inputs.                 */
/* PARAMETER USAGE :                                           */
/*    Uses in Argv[1...3]                                      */
/* FUNCTION CALLED :                                           */
/*    N/A                                                      */
/* ----------------------------------------------------------- */
int main(int argc, char *argv[]) {
        //Initial Argument Checking
        if (argc != 4) {
                printf("ERROR: ./prog4 m n k\n");
                return -1;
        }

        int m = atoi(argv[1]);
        int n = atoi(argv[2]);
        int k = atoi(argv[3]);

        if (m == 0) {
                m = 5;
        }

        if (n == 0) {
                n = 5;
        }

        if (k == 0) {
                k = 10;
        }

        //Spawns the landlord and student Threads
        spawnThreads(m, n, k);
        return 1;
}
```

```
/* ----------------------------------------------------------- */
/* NAME : Bryan Wandrych                     User ID: bdwandry */
/* DUE DATE : 12/4/2021                                        */
/* PROGRAM ASSIGNMENT 4                                        */
/* FILE NAME : thread.h                                        */
/* PROGRAM PURPOSE : The purpose of this program is to solve   */
/* the Party Problem by implementing 2 different types of      */
/* threads, students and landlord. Students                    */
/* will join/leave a room freely. However, if the landlord     */
/* appears at a random time and the room is over capacity, they*/
/* will remove everyone from that room. This files function is */
/* to create a link and shared memory between the threads.     *.
/* As wellas creating a link to threadmentor in the backend.   */
/* ----------------------------------------------------------- */

#include "ThreadClass.h"

//Global variables that share values between Main and thread.cpp class
extern Semaphore ExSem;
extern Semaphore EntSem;
extern Mutex PtrLck;

/* ----------------------------------------------------------- */
/* FUNCTION: StudentsThread (Class Definition)                 */
/*    This is the header file that describes the threadmentor's*/
/*        class. This class will be shared with all files when     */
/*        threads are called for student based threads.            */
/* PARAMETER USAGE :                                           */
/*    int i - used to pass through the iteration index         */
/* FUNCTION CALLED :                                           */
/*    thread.cpp                                               */
/*    Threadmentor                                             */
/* ----------------------------------------------------------- */
class StudentsThread : public Thread {
    public:
        StudentsThread(int i);
                static int studentsInRoom;
                static int totalProcesses;

    private:
                int i; int studentID;
                int GoToParty(); void ThreadFunc();

};

/* ----------------------------------------------------------- */
/* FUNCTION: LandLordThread (Class Definition)                 */
/*    This is the header file that describes the threadmentor's*/
/*        class. This class will be shared with all files when     */
/*        threads are called for student based threads.            */
/* PARAMETER USAGE :                                           */
/*    int m - used for number of times the landlord checks the */
/*            room                                             */
/*    int n - Keeps track of the rooms capacity                */
/* FUNCTION CALLED :                                           */
/*    thread.cpp                                               */
/*    Threadmentor                                             */
/* ----------------------------------------------------------- */
class LandLordThread : public Thread {
    public:
        LandLordThread(int m, int n);
                static int completed;

    private:
                int m; int n; int numOfAptmSearches; int capacity; int iteration;
                int CheckRoom(); void ThreadFunc();
};
```

```
/* ----------------------------------------------------------- */
/* NAME : Bryan Wandrych                     User ID: bdwandry */
/* DUE DATE : 12/4/2021                                        */
/* PROGRAM ASSIGNMENT 4                                        */
/* FILE NAME : thread.h                                        */
/* PROGRAM PURPOSE : The purpose of this program is to solve   */
/* the Party Problem by implementing 2 different types of      */
/* threads, students and landlord. Students                    */
/* will join/leave a room freely. However, if the landlord     */
/* appears at a random time and the room is over capacity, they*/
/* will remove everyone from that room. This file contains     */
/* starting logic for both the Students and the Landlord.      */
/* ----------------------------------------------------------- */

#include <iostream>
#include "thread.h"
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
using namespace std;

int StudentsThread::studentsInRoom = 0;
int StudentsThread::totalProcesses = 0;
int LandLordThread::completed = 0;
Semaphore ExSem("Exit", 1);
Semaphore EntSem("Enter", 1);
Mutex PtrLck("Printing");

/* ----------------------------------------------------------- */
/* FUNCTION: StudentsThread                                    */
/*    This is the constructor to the StudentsThread class      */
/*    described in the thread.h. It will supply definitions    */
/*    for pass through arguments passed from main.             */
/* PARAMETER USAGE :                                           */
/*    int i = Passed from Main of the index for each           */
/*        specificfied student thread.                             */
/* FUNCTION CALLED:                                            */
/*    ThreadMentor                                             */
/* ----------------------------------------------------------- */
StudentsThread::StudentsThread(int i) {
        studentID = i;
}

/* ----------------------------------------------------------- */
/* FUNCTION: LandLordThread                                    */
/*    This is the constructor to the LandLordThread class      */
/*    described in the thread.h. It will supply definitions    */
/*    for pass through arguments passed from main.             */
/* PARAMETER USAGE :                                           */
/*    int m - used for number of times the landlord checks the */
/*            room                                             */
/*    int n - Keeps track of the rooms capacity                */
/* FUNCTION CALLED :                                           */
/*    Threadmentor                                             */
/* ----------------------------------------------------------- */
LandLordThread::LandLordThread(int m, int n) {
        numOfAptmSearches = m;
        capacity = n;
        iteration = 0;
}

/* ----------------------------------------------------------- */
/* FUNCTION: Student ThreadFunc                                */
```

```
/*    This is the executing code for the specific thread       */
/*        created. It will call the GoToParty function for the     */
/*    Students to enter in the room                            */
/* PARAMETER USAGE :                                           */
/*    N/A                                                      */
/* FUNCTION CALLED :                                           */
/*    ThreadMentor                                             */
/*    Main                                                     */
/* ----------------------------------------------------------- */
void StudentsThread::ThreadFunc() {
        Thread::ThreadFunc();
        PtrLck.Lock();
                printf("     Student %d starts\n", studentID);
        PtrLck.Unlock();
        while(LandLordThread::completed != 1) {
                Delay();
                GoToParty();
                Delay();
        }

        PtrLck.Lock();
                printf("     Student %d terminates\n", studentID + 1);
                StudentsThread::totalProcesses--;
        PtrLck.Unlock();
        Exit();
}

/* ----------------------------------------------------------- */
/* FUNCTION: Landlord ThreadFunc                               */
/*    This is the executing code for the specific thread       */
/*        created. It will call the CheckRoom function for the     */
/*    Landlord to enter in the room and see if its over        */
/*    capacity                                                 */
/* PARAMETER USAGE :                                           */
/*    N/A                                                      */
/* FUNCTION CALLED :                                           */
/*    ThreadMentor                                             */
/*    Main                                                     */
/* ----------------------------------------------------------- */
void LandLordThread::ThreadFunc() {
        Thread::ThreadFunc();
        for (int i = 0; i < numOfAptmSearches; i++) {
                iteration = i;
                Delay();
                CheckRoom();
                //printf("Landlord: %d/%d; Capacity: %d\n", i+1, numOfAptmSearches,
capacity);
                if ((i + 1) == numOfAptmSearches) {
                        LandLordThread::completed = 1;
                        while(StudentsThread::totalProcesses != 0) {
                                Delay();
                        }
                        break;
                }
                Delay();
        }

        PtrLck.Lock();
                printf("After checking the room %d times, the landlord retires and is on
vacation!\n", numOfAptmSearches);
        PtrLck.Unlock();

        Exit();
}
```

```
/* ----------------------------------------------------------- */
/* NAME : Bryan Wandrych                     User ID: bdwandry */
/* DUE DATE : 12/4/2021                                        */
/* PROGRAM ASSIGNMENT 4                                        */
/* FILE NAME : thread.h                                        */
/* PROGRAM PURPOSE : The purpose of this program is to solve   */
/* the Party Problem by implementing 2 different types of      */
/* threads, students and landlord. Students                    */
/* will join/leave a room freely. However, if the landlord     */
/* appears at a random time and the room is over capacity, they*/
/* will remove everyone from that room. This file contains     */
/* all of the logic to the Students and Landlord.              */
/* ----------------------------------------------------------- */
#include <iostream>
#include "thread.h"
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
using namespace std;

/* ----------------------------------------------------------- */
/* FUNCTION: GoToParty                                         */
/*    This is an extension of Thread.cpp. It will aid in the   */
/*    logic behind the Student threads generated by this       */
/*    algorithm.                                                   */
/* PARAMETER USAGE :                                           */
/*    N/A                                                      */
/* FUNCTION CALLED :                                           */
/*    StudentsThread                                           */
/* ----------------------------------------------------------- */
int StudentsThread::GoToParty() {
        //------------------------------------------------------------------
        //Enters Room
        //Mutex locking: If a student wants to enter the room.
        Delay();
        PtrLck.Lock();
                printf("     Student %d waits to enter the room\n",  studentID + 1);
        PtrLck.Unlock();

        //Adds some randomized delay
        for (int i = 0; i < 5; i++) {
                Delay();
        }
        //Mutex Locking: The student tries to enters the room.
        EntSem.Wait();
                PtrLck.Lock();
                        StudentsThread::studentsInRoom++;
                        printf("     Student %d enters the room (%d students in the
room) \n",  studentID + 1, StudentsThread::studentsInRoom);
                PtrLck.Unlock();
        EntSem.Signal();
        //Adds some randomized delay
        for (int i = 0; i < (StudentsThread::studentsInRoom); i++) {
                Delay();
        }
        //------------------------------------------------------------------
        //Leaves Room
        //Mutex Locking: The student is getting ready to leave the room.
        PtrLck.Lock();
                printf("     Student %d waits to leave the room\n", studentID + 1);
        PtrLck.Unlock();

        //Mutex Locking: The student is getting ready to leave the room.
```

```
        ExSem.Wait();
                PtrLck.Lock();
                        StudentsThread::studentsInRoom--;
                        printf("     Student %d leaves the room (%d students in the
room)\n",  studentID + 1, StudentsThread::studentsInRoom);
                PtrLck.Unlock();
        ExSem.Signal();
        return 1;
}

/* ----------------------------------------------------------- */
/* FUNCTION: CheckRoom                                         */
/*    This is an extension of Thread.cpp. It will aid in the   */
/*    logic behind the Landlord threads generated by this      */
/*    algorithm.                                                   */
/* PARAMETER USAGE :                                           */
/*    N/A                                                      */
/* FUNCTION CALLED :                                           */
/*    LandLordThread                                           */
/* ----------------------------------------------------------- */
int LandLordThread::CheckRoom() {
        //Mutex Lock: Landlord has officially Entered the room
        //Blocks all students from entering and leaving.
        EntSem.Wait();
        ExSem.Wait();

        PtrLck.Lock();
                        printf("The landlord checks the room the %d time\n",  iteration
+ 1);
                        //printf("Number of Current Students: %d; Capacity of Room:
%d\n", StudentsThread::studentsInRoom, capacity);
        PtrLck.Unlock();

        //Landlord Enters - Checks to see if the room is empty
        if (StudentsThread::studentsInRoom == 0) {
                PtrLck.Lock();
                        printf("The landlord finds the room has no students and
leaves\n");
                PtrLck.Unlock();
                EntSem.Signal();
                ExSem.Signal();
        }

        //Landlord Enters - Checks to see if there are more people greater than the
given capacity (i.e. n >= k)
        else if (StudentsThread::studentsInRoom >= capacity) {
                PtrLck.Lock();
                        printf("The landlord finds %d students in the room and breaks up
the party\n", StudentsThread::studentsInRoom);
                PtrLck.Unlock();
                PtrLck.Lock();
                        printf("The landlord finishes checking and forces everyone to
leave\n");
                PtrLck.Unlock();
                ExSem.Signal();
                while(StudentsThread::studentsInRoom != 0) {
                        Delay();
                }
                PtrLck.Lock();
                        printf("The landlord leaves the room and the room is empty\n");
                PtrLck.Unlock();
                EntSem.Signal();
        }
        //Landlord Enters - Checks to see if there are less people than the given
```

```
capacity (i.e. n < k)
        else if (StudentsThread::studentsInRoom < capacity) {
                PtrLck.Lock();
                        printf("The landlord finds there are %d students in the room
(condition being good) and leaves.\n", StudentsThread::studentsInRoom);
                PtrLck.Unlock();
                EntSem.Signal();
                ExSem.Signal();
        }
        return 1;
}
```

```
CC       = c++
FLAGS    =
CFLAGS   = -g -O2 -Wno-write-strings -Wno-cpp -w
DFLAGS   = -DPACKAGE=\"threadsystem\" -DVERSION=\"1.0\" -DPTHREAD=1 -DUNIX_MSG_Q=1
-DSTDC_HEADERS=1
IFLAGS   = -I/local/eit-linux/apps/ThreadMentor/include
TMLIB    = /local/eit-linux/apps/ThreadMentor/Visual/libthreadclass.a
TMLIB_NV    = /local/eit-linux/apps/ThreadMentor/NoVisual/libthreadclass.a
OBJ_FILE = thread.o thread-main.o thread-support.o
EXE_FILE = prog4
${EXE_FILE}: ${OBJ_FILE}
                ${CC} ${FLAGS}  -o ${EXE_FILE}  ${OBJ_FILE} ${TMLIB_NV} -lpthread

thread.o: thread.cpp
                ${CC} ${DFLAGS} ${IFLAGS} ${CFLAGS} -c thread.cpp

thread-main.o: thread-main.cpp
                ${CC} ${DFLAGS} ${IFLAGS} ${CFLAGS} -c thread-main.cpp

thread-support.o: thread-support.cpp
                ${CC} ${DFLAGS} ${IFLAGS} ${CFLAGS} -c thread-support.cpp

noVisual: ${OBJ_FILE}
                ${CC} ${FLAGS}  -o ${EXE_FILE}  ${OBJ_FILE} ${TMLIB_NV} -lpthread

clean:
                rm -f ${OBJ_FILE} ${EXE_FILE}
```

```
========================== COMPILATION ==========================
rm -f thread.o thread-main.o thread-support.o prog4
c++ -DPACKAGE=\"threadsystem\" -DVERSION=\"1.0\" -DPTHREAD=1 -DUNIX_MSG_Q=1
-DSTDC_HEADERS=1 -I/local/eit-linux/apps/ThreadMentor/include -g -O2 -Wno-write-strings
-Wno-cpp -w -c thread.cpp
c++ -DPACKAGE=\"threadsystem\" -DVERSION=\"1.0\" -DPTHREAD=1 -DUNIX_MSG_Q=1
-DSTDC_HEADERS=1 -I/local/eit-linux/apps/ThreadMentor/include -g -O2 -Wno-write-strings
-Wno-cpp -w -c thread-main.cpp
c++ -DPACKAGE=\"threadsystem\" -DVERSION=\"1.0\" -DPTHREAD=1 -DUNIX_MSG_Q=1
-DSTDC_HEADERS=1 -I/local/eit-linux/apps/ThreadMentor/include -g -O2 -Wno-write-strings
-Wno-cpp -w -c thread-support.cpp
c++   -o prog4  thread.o thread-main.o thread-support.o
/local/eit-linux/apps/ThreadMentor/NoVisual/libthreadclass.a -lpthread
Compilation done.
=========================== TEST 1 ===========================
The landlord checks the room the 1 time
     Student 4 starts
     Student 0 starts
     Student 8 starts
     Student 2 starts
     Student 12 starts
The landlord finds the room has no students and leaves
     Student 1 starts
The landlord checks the room the 2 time
The landlord finds the room has no students and leaves
     Student 7 starts
     Student 2 waits to enter the room
     Student 9 starts
     Student 1 waits to enter the room
     Student 3 starts
     Student 3 waits to enter the room
     Student 4 waits to enter the room
     Student 13 starts
     Student 9 waits to enter the room
     Student 11 starts
     Student 14 starts
     Student 12 waits to enter the room
     Student 15 starts
     Student 15 waits to enter the room
     Student 16 waits to enter the room
     Student 5 starts
     Student 6 starts
     Student 17 starts
     Student 7 waits to enter the room
     Student 6 waits to enter the room
The landlord checks the room the 3 time
The landlord finds the room has no students and leaves
     Student 18 starts
     Student 2 enters the room (1 students in the room)
     Student 18 waits to enter the room
     Student 19 waits to enter the room
     Student 14 waits to enter the room
     Student 10 starts
     Student 13 waits to enter the room
     Student 16 starts
     Student 11 waits to enter the room
     Student 8 waits to enter the room
     Student 17 waits to enter the room
     Student 5 waits to enter the room
     Student 2 waits to leave the room
     Student 2 leaves the room (0 students in the room)
     Student 16 enters the room (1 students in the room)
     Student 19 starts
     Student 17 enters the room (2 students in the room)
     Student 20 waits to enter the room
     Student 5 enters the room (3 students in the room)
     Student 16 waits to leave the room
     Student 17 waits to leave the room
     Student 2 waits to enter the room
     Student 10 waits to enter the room
     Student 16 leaves the room (2 students in the room)
     Student 3 enters the room (3 students in the room)
     Student 5 waits to leave the room
     Student 20 enters the room (4 students in the room)
     Student 10 enters the room (5 students in the room)
     Student 16 waits to enter the room
     Student 3 waits to leave the room
     Student 17 leaves the room (4 students in the room)
     Student 20 waits to leave the room
     Student 5 leaves the room (3 students in the room)
The landlord checks the room the 4 time
The landlord finds there are 3 students in the room (condition being good) and leaves.
     Student 10 waits to leave the room
     Student 1 enters the room (4 students in the room)
     Student 17 waits to enter the room
     Student 5 waits to enter the room
     Student 3 leaves the room (3 students in the room)
     Student 20 leaves the room (2 students in the room)
     Student 3 waits to enter the room
     Student 10 leaves the room (1 students in the room)
     Student 1 waits to leave the room
The landlord checks the room the 5 time
The landlord finds there are 1 students in the room (condition being good) and leaves.
     Student 7 enters the room (2 students in the room)
     Student 20 waits to enter the room
     Student 10 waits to enter the room
     Student 1 leaves the room (1 students in the room)
     Student 7 waits to leave the room
     Student 7 leaves the room (0 students in the room)
     Student 18 enters the room (1 students in the room)
     Student 14 enters the room (2 students in the room)
     Student 1 waits to enter the room
     Student 7 waits to enter the room
     Student 18 waits to leave the room
     Student 14 waits to leave the room
     Student 19 enters the room (3 students in the room)
     Student 18 leaves the room (2 students in the room)
     Student 11 enters the room (3 students in the room)
     Student 14 leaves the room (2 students in the room)
     Student 19 waits to leave the room
     Student 19 leaves the room (1 students in the room)
     Student 18 waits to enter the room
     Student 14 waits to enter the room
     Student 13 enters the room (2 students in the room)
     Student 11 waits to leave the room
     Student 11 leaves the room (1 students in the room)
     Student 19 waits to enter the room
     Student 8 enters the room (2 students in the room)
     Student 11 waits to enter the room
     Student 4 enters the room (3 students in the room)
     Student 13 waits to leave the room
     Student 8 waits to leave the room
     Student 9 enters the room (4 students in the room)
     Student 13 leaves the room (3 students in the room)
     Student 4 waits to leave the room
     Student 4 leaves the room (2 students in the room)
     Student 12 enters the room (3 students in the room)
     Student 8 leaves the room (2 students in the room)
     Student 2 enters the room (3 students in the room)
     Student 9 waits to leave the room
     Student 13 waits to enter the room
     Student 9 leaves the room (2 students in the room)
     Student 4 waits to enter the room
     Student 8 waits to enter the room
     Student 12 waits to leave the room
     Student 15 enters the room (3 students in the room)
     Student 2 waits to leave the room
     Student 12 leaves the room (2 students in the room)
     Student 15 waits to leave the room
     Student 2 leaves the room (1 students in the room)
     Student 16 enters the room (2 students in the room)
     Student 15 leaves the room (1 students in the room)
     Student 2 waits to enter the room
     Student 6 enters the room (2 students in the room)
     Student 16 waits to leave the room
     Student 12 waits to enter the room
     Student 16 leaves the room (1 students in the room)
     Student 9 waits to enter the room
     Student 17 enters the room (2 students in the room)
     Student 6 waits to leave the room
     Student 16 waits to enter the room
     Student 6 leaves the room (1 students in the room)
     Student 15 waits to enter the room
     Student 17 waits to leave the room
     Student 2 enters the room (2 students in the room)
     Student 9 enters the room (3 students in the room)
     Student 17 leaves the room (2 students in the room)
     Student 9 waits to leave the room
     Student 9 leaves the room (1 students in the room)
     Student 20 enters the room (2 students in the room)
     Student 6 waits to enter the room
     Student 9 waits to enter the room
     Student 2 waits to leave the room
     Student 20 waits to leave the room
     Student 1 enters the room (3 students in the room)
     Student 2 leaves the room (2 students in the room)
     Student 17 waits to enter the room
     Student 9 enters the room (3 students in the room)
     Student 2 waits to enter the room
     Student 20 leaves the room (2 students in the room)
     Student 10 enters the room (3 students in the room)
     Student 9 waits to leave the room
     Student 1 waits to leave the room
     Student 9 leaves the room (2 students in the room)
     Student 7 enters the room (3 students in the room)
     Student 1 leaves the room (2 students in the room)
     Student 10 waits to leave the room
     Student 14 enters the room (3 students in the room)
     Student 10 leaves the room (2 students in the room)
     Student 9 waits to enter the room
     Student 7 waits to leave the room
     Student 20 waits to enter the room
     Student 14 waits to leave the room
     Student 1 waits to enter the room
     Student 7 leaves the room (1 students in the room)
     Student 2 enters the room (2 students in the room)
     Student 14 leaves the room (1 students in the room)
     Student 10 waits to enter the room
     Student 1 enters the room (2 students in the room)
     Student 7 waits to enter the room
```

```
     Student 14 waits to enter the room
     Student 1 waits to leave the room
     Student 1 leaves the room (1 students in the room)
     Student 2 waits to leave the room
     Student 2 leaves the room (0 students in the room)
     Student 19 enters the room (1 students in the room)
     Student 1 waits to enter the room
     Student 8 enters the room (2 students in the room)
     Student 1 enters the room (3 students in the room)
     Student 2 waits to enter the room
     Student 13 enters the room (4 students in the room)
     Student 19 waits to leave the room
     Student 19 leaves the room (3 students in the room)
     Student 1 waits to leave the room
     Student 5 enters the room (4 students in the room)
     Student 8 waits to leave the room
     Student 8 leaves the room (3 students in the room)
     Student 1 leaves the room (2 students in the room)
     Student 19 waits to enter the room
     Student 12 enters the room (3 students in the room)
     Student 13 waits to leave the room
The landlord checks the room the 6 time
The landlord finds there are 3 students in the room (condition being good) and leaves.
     Student 1 waits to enter the room
     Student 8 waits to enter the room
     Student 5 waits to leave the room
     Student 13 leaves the room (2 students in the room)
     Student 15 enters the room (3 students in the room)
     Student 12 waits to leave the room
     Student 1 enters the room (4 students in the room)
     Student 12 leaves the room (3 students in the room)
     Student 13 waits to enter the room
     Student 3 enters the room (4 students in the room)
     Student 5 leaves the room (3 students in the room)
     Student 15 waits to leave the room
     Student 1 waits to leave the room
     Student 3 waits to leave the room
     Student 15 leaves the room (2 students in the room)
     Student 6 enters the room (3 students in the room)
     Student 3 leaves the room (2 students in the room)
     Student 13 enters the room (3 students in the room)
     Student 5 waits to enter the room
     Student 1 leaves the room (2 students in the room)
     Student 12 waits to enter the room
     Student 13 waits to leave the room
     Student 3 waits to enter the room
     Student 18 enters the room (3 students in the room)
     Student 15 waits to enter the room
     Student 13 leaves the room (2 students in the room)
     Student 1 waits to enter the room
     Student 6 waits to leave the room
     Student 9 enters the room (3 students in the room)
     Student 6 leaves the room (2 students in the room)
     Student 18 waits to leave the room
     Student 12 enters the room (3 students in the room)
     Student 9 waits to leave the room
     Student 9 leaves the room (2 students in the room)
     Student 13 waits to enter the room
     Student 10 enters the room (3 students in the room)
     Student 18 leaves the room (2 students in the room)
     Student 6 waits to enter the room
     Student 9 waits to enter the room
     Student 12 waits to leave the room
     Student 20 enters the room (3 students in the room)
     Student 10 waits to leave the room
     Student 12 leaves the room (2 students in the room)
     Student 18 waits to enter the room
     Student 10 leaves the room (1 students in the room)
     Student 14 enters the room (2 students in the room)
     Student 20 waits to leave the room
     Student 7 enters the room (3 students in the room)
     Student 12 waits to enter the room
     Student 4 enters the room (4 students in the room)
     Student 20 leaves the room (3 students in the room)
     Student 7 waits to leave the room
     Student 2 enters the room (4 students in the room)
     Student 7 leaves the room (3 students in the room)
     Student 14 waits to leave the room
     Student 10 waits to enter the room
     Student 20 waits to enter the room
     Student 2 waits to leave the room
     Student 4 waits to leave the room
     Student 14 leaves the room (2 students in the room)
     Student 4 leaves the room (1 students in the room)
     Student 18 enters the room (2 students in the room)
     Student 7 waits to enter the room
     Student 2 leaves the room (1 students in the room)
     Student 14 waits to enter the room
     Student 20 enters the room (2 students in the room)
     Student 18 waits to leave the room
     Student 4 waits to enter the room
     Student 20 waits to leave the room
     Student 18 leaves the room (1 students in the room)
     Student 2 waits to enter the room
     Student 20 leaves the room (0 students in the room)
     Student 16 enters the room (1 students in the room)
     Student 18 waits to enter the room
     Student 20 waits to enter the room
     Student 2 enters the room (2 students in the room)
     Student 16 waits to leave the room
     Student 2 waits to leave the room
     Student 17 enters the room (3 students in the room)
     Student 20 enters the room (4 students in the room)
     Student 16 leaves the room (3 students in the room)
     Student 11 enters the room (4 students in the room)
     Student 2 leaves the room (3 students in the room)
     Student 15 enters the room (4 students in the room)
     Student 17 waits to leave the room
     Student 2 waits to enter the room
     Student 11 waits to leave the room
     Student 5 enters the room (5 students in the room)
     Student 2 enters the room (6 students in the room)
     Student 16 waits to enter the room
     Student 1 enters the room (7 students in the room)
     Student 20 waits to leave the room
     Student 15 waits to leave the room
     Student 6 enters the room (8 students in the room)
     Student 17 leaves the room (7 students in the room)
     Student 5 waits to leave the room
     Student 5 leaves the room (6 students in the room)
     Student 2 waits to leave the room
     Student 9 enters the room (7 students in the room)
     Student 20 leaves the room (6 students in the room)
     Student 17 waits to enter the room
     Student 1 waits to leave the room
     Student 5 waits to enter the room
The landlord checks the room the 7 time
The landlord finds 6 students in the room and breaks up the party
```

```
The landlord finishes checking and forces everyone to leave
     Student 11 leaves the room (5 students in the room)
     Student 9 waits to leave the room
     Student 9 leaves the room (4 students in the room)
     Student 2 leaves the room (3 students in the room)
     Student 20 waits to enter the room
     Student 15 leaves the room (2 students in the room)
     Student 6 waits to leave the room
     Student 1 leaves the room (1 students in the room)
     Student 9 waits to enter the room
     Student 6 leaves the room (0 students in the room)
     Student 11 waits to enter the room
     Student 2 waits to enter the room
     Student 15 waits to enter the room
     Student 1 waits to enter the room
     Student 6 waits to enter the room
The landlord leaves the room and the room is empty
     Student 12 enters the room (1 students in the room)
     Student 9 enters the room (2 students in the room)
     Student 15 enters the room (3 students in the room)
     Student 8 enters the room (4 students in the room)
     Student 10 enters the room (5 students in the room)
     Student 7 enters the room (6 students in the room)
     Student 1 enters the room (7 students in the room)
     Student 4 enters the room (8 students in the room)
     Student 19 enters the room (9 students in the room)
     Student 8 waits to leave the room
     Student 8 leaves the room (8 students in the room)
     Student 3 enters the room (9 students in the room)
     Student 13 enters the room (10 students in the room)
     Student 15 waits to leave the room
     Student 8 waits to enter the room
     Student 15 leaves the room (9 students in the room)
     Student 17 enters the room (10 students in the room)
     Student 8 enters the room (11 students in the room)
     Student 5 enters the room (12 students in the room)
     Student 20 enters the room (13 students in the room)
     Student 15 waits to enter the room
The landlord checks the room the 8 time
The landlord finds 13 students in the room and breaks up the party
The landlord finishes checking and forces everyone to leave
     Student 7 waits to leave the room
     Student 7 leaves the room (12 students in the room)
     Student 12 waits to leave the room
     Student 9 waits to leave the room
     Student 8 waits to leave the room
     Student 7 waits to enter the room
     Student 12 leaves the room (11 students in the room)
     Student 3 waits to leave the room
     Student 13 waits to leave the room
     Student 3 leaves the room (10 students in the room)
     Student 5 waits to leave the room
     Student 1 waits to leave the room
     Student 20 waits to leave the room
     Student 13 leaves the room (9 students in the room)
     Student 17 waits to leave the room
     Student 8 leaves the room (8 students in the room)
     Student 5 leaves the room (7 students in the room)
     Student 10 waits to leave the room
     Student 5 waits to enter the room
     Student 19 waits to leave the room
     Student 4 waits to leave the room
     Student 12 waits to enter the room
     Student 13 waits to enter the room
     Student 3 waits to enter the room
     Student 8 waits to enter the room
     Student 20 leaves the room (6 students in the room)
     Student 1 leaves the room (5 students in the room)
     Student 9 leaves the room (4 students in the room)
     Student 17 leaves the room (3 students in the room)
     Student 1 waits to enter the room
     Student 10 leaves the room (2 students in the room)
     Student 9 waits to enter the room
     Student 20 waits to enter the room
     Student 10 waits to enter the room
     Student 17 waits to enter the room
     Student 19 leaves the room (1 students in the room)
     Student 4 leaves the room (0 students in the room)
The landlord leaves the room and the room is empty
     Student 19 waits to enter the room
     Student 11 enters the room (1 students in the room)
     Student 4 waits to enter the room
     Student 6 enters the room (2 students in the room)
     Student 14 enters the room (3 students in the room)
     Student 11 waits to leave the room
     Student 14 waits to leave the room
     Student 11 leaves the room (2 students in the room)
     Student 6 waits to leave the room
     Student 18 enters the room (3 students in the room)
     Student 14 leaves the room (2 students in the room)
     Student 11 terminates
     Student 18 waits to leave the room
     Student 14 terminates
     Student 16 enters the room (3 students in the room)
     Student 6 leaves the room (2 students in the room)
     Student 2 enters the room (3 students in the room)
     Student 18 leaves the room (2 students in the room)
     Student 2 waits to leave the room
     Student 2 leaves the room (1 students in the room)
     Student 2 terminates
     Student 15 enters the room (2 students in the room)
     Student 18 terminates
     Student 6 terminates
     Student 7 enters the room (3 students in the room)
     Student 16 waits to leave the room
     Student 15 waits to leave the room
     Student 5 enters the room (4 students in the room)
     Student 16 leaves the room (3 students in the room)
     Student 8 enters the room (4 students in the room)
     Student 15 leaves the room (3 students in the room)
     Student 3 enters the room (4 students in the room)
     Student 8 waits to leave the room
     Student 16 terminates
     Student 8 leaves the room (3 students in the room)
     Student 7 waits to leave the room
     Student 13 enters the room (4 students in the room)
     Student 7 leaves the room (3 students in the room)
     Student 15 terminates
     Student 7 terminates
     Student 13 waits to leave the room
     Student 13 leaves the room (2 students in the room)
     Student 8 terminates
     Student 3 waits to leave the room
     Student 3 leaves the room (1 students in the room)
     Student 5 waits to leave the room
     Student 13 terminates
     Student 3 terminates
     Student 5 leaves the room (0 students in the room)
```

```
     Student 12 enters the room (1 students in the room)
     Student 5 terminates
     Student 1 enters the room (2 students in the room)
     Student 10 enters the room (3 students in the room)
     Student 12 waits to leave the room
     Student 9 enters the room (4 students in the room)
     Student 10 waits to leave the room
     Student 12 leaves the room (3 students in the room)
     Student 1 waits to leave the room
     Student 1 leaves the room (2 students in the room)
     Student 20 enters the room (3 students in the room)
     Student 10 leaves the room (2 students in the room)
     Student 12 terminates
     Student 10 terminates
     Student 17 enters the room (3 students in the room)
     Student 1 terminates
     Student 4 enters the room (4 students in the room)
     Student 9 waits to leave the room
     Student 20 waits to leave the room
     Student 19 enters the room (5 students in the room)
     Student 17 waits to leave the room
     Student 9 leaves the room (4 students in the room)
     Student 9 terminates
     Student 20 leaves the room (3 students in the room)
     Student 4 waits to leave the room
     Student 4 leaves the room (2 students in the room)
     Student 17 leaves the room (1 students in the room)
     Student 17 terminates
     Student 20 terminates
     Student 19 waits to leave the room
     Student 19 leaves the room (0 students in the room)
     Student 4 terminates
     Student 19 terminates
After checking the room 8 times, the landlord retires and is on vacation!

=========================== TEST 2 ===========================
     Student 0 starts
     Student 5 starts
The landlord checks the room the 1 time
     Student 1 starts
     Student 2 starts
     Student 3 starts
     Student 4 starts
     Student 14 starts
     Student 6 starts
     Student 15 waits to enter the room
     Student 5 waits to enter the room
     Student 7 starts
     Student 8 starts
     Student 6 waits to enter the room
The landlord finds the room has no students and leaves
     Student 9 starts
     Student 10 starts
     Student 2 waits to enter the room
     Student 11 starts
     Student 3 waits to enter the room
     Student 10 waits to enter the room
     Student 11 waits to enter the room
     Student 4 waits to enter the room
     Student 1 waits to enter the room
     Student 7 waits to enter the room
     Student 5 enters the room (1 students in the room)
     Student 3 enters the room (2 students in the room)
     Student 9 waits to enter the room
     Student 1 enters the room (3 students in the room)
     Student 12 waits to enter the room
     Student 6 enters the room (4 students in the room)
     Student 12 starts
     Student 2 enters the room (5 students in the room)
     Student 3 waits to leave the room
     Student 12 enters the room (6 students in the room)
     Student 13 starts
     Student 3 leaves the room (5 students in the room)
     Student 5 waits to leave the room
     Student 10 enters the room (6 students in the room)
     Student 5 leaves the room (5 students in the room)
     Student 1 waits to leave the room
     Student 4 enters the room (6 students in the room)
     Student 8 waits to enter the room
     Student 1 leaves the room (5 students in the room)
     Student 13 waits to enter the room
     Student 3 waits to enter the room
     Student 5 waits to enter the room
     Student 2 waits to leave the room
     Student 14 waits to enter the room
     Student 12 waits to leave the room
     Student 10 waits to leave the room
     Student 6 waits to leave the room
The landlord checks the room the 2 time
The landlord finds there are 5 students in the room (condition being good) and leaves.
     Student 8 enters the room (6 students in the room)
     Student 4 waits to leave the room
     Student 4 leaves the room (5 students in the room)
     Student 1 waits to enter the room
     Student 12 leaves the room (4 students in the room)
     Student 8 waits to leave the room
     Student 10 leaves the room (3 students in the room)
     Student 15 enters the room (4 students in the room)
     Student 6 leaves the room (3 students in the room)
     Student 4 waits to enter the room
     Student 12 waits to enter the room
     Student 9 enters the room (4 students in the room)
     Student 10 waits to enter the room
     Student 2 leaves the room (3 students in the room)
     Student 15 waits to leave the room
     Student 8 leaves the room (2 students in the room)
     Student 1 enters the room (3 students in the room)
     Student 2 waits to enter the room
     Student 8 waits to enter the room
     Student 15 leaves the room (2 students in the room)
     Student 9 waits to leave the room
     Student 9 leaves the room (1 students in the room)
     Student 6 waits to enter the room
     Student 15 waits to enter the room
     Student 13 enters the room (2 students in the room)
     Student 9 waits to enter the room
     Student 3 enters the room (3 students in the room)
     Student 1 waits to leave the room
     Student 2 enters the room (4 students in the room)
     Student 5 enters the room (5 students in the room)
     Student 1 leaves the room (4 students in the room)
     Student 3 waits to leave the room
     Student 7 enters the room (5 students in the room)
     Student 3 leaves the room (4 students in the room)
     Student 13 waits to leave the room
     Student 2 waits to leave the room
     Student 5 waits to leave the room
     Student 2 leaves the room (3 students in the room)
```

```
     Student 3 waits to enter the room
     Student 15 enters the room (4 students in the room)
     Student 1 waits to enter the room
     Student 7 waits to leave the room
     Student 7 leaves the room (3 students in the room)
     Student 3 enters the room (4 students in the room)
     Student 13 leaves the room (3 students in the room)
     Student 15 waits to leave the room
     Student 12 enters the room (4 students in the room)
     Student 3 waits to leave the room
     Student 2 waits to enter the room
     Student 15 leaves the room (3 students in the room)
     Student 4 enters the room (4 students in the room)
     Student 13 waits to enter the room
     Student 5 leaves the room (3 students in the room)
     Student 10 enters the room (4 students in the room)
     Student 7 waits to enter the room
     Student 12 waits to leave the room
     Student 12 leaves the room (3 students in the room)
     Student 10 waits to leave the room
     Student 3 leaves the room (2 students in the room)
     Student 15 waits to enter the room
     Student 5 waits to enter the room
     Student 8 enters the room (3 students in the room)
     Student 12 waits to enter the room
     Student 4 waits to leave the room
     Student 10 leaves the room (2 students in the room)
     Student 3 waits to enter the room
     Student 8 waits to leave the room
     Student 4 leaves the room (1 students in the room)
     Student 10 waits to enter the room
     Student 14 enters the room (2 students in the room)
     Student 8 leaves the room (1 students in the room)
     Student 6 enters the room (2 students in the room)
     Student 6 waits to leave the room
     Student 14 waits to leave the room
     Student 4 waits to enter the room
     Student 6 leaves the room (1 students in the room)
     Student 9 enters the room (2 students in the room)
     Student 14 leaves the room (1 students in the room)
     Student 8 waits to enter the room
     Student 9 waits to leave the room
     Student 9 leaves the room (0 students in the room)
The landlord checks the room the 3 time
The landlord finds the room has no students and leaves
     Student 6 waits to enter the room
     Student 11 enters the room (1 students in the room)
     Student 9 waits to enter the room
     Student 14 waits to enter the room
     Student 1 enters the room (2 students in the room)
     Student 11 waits to leave the room
     Student 11 leaves the room (1 students in the room)
     Student 8 enters the room (2 students in the room)
     Student 1 waits to leave the room
     Student 1 leaves the room (1 students in the room)
     Student 13 enters the room (2 students in the room)
     Student 11 waits to enter the room
     Student 7 enters the room (3 students in the room)
     Student 8 waits to leave the room
     Student 8 leaves the room (2 students in the room)
     Student 1 waits to enter the room
     Student 13 waits to leave the room
     Student 15 enters the room (3 students in the room)
     Student 13 leaves the room (2 students in the room)
     Student 8 waits to enter the room
     Student 12 enters the room (3 students in the room)
     Student 7 waits to leave the room
     Student 7 leaves the room (2 students in the room)
     Student 13 waits to enter the room
     Student 3 enters the room (3 students in the room)
     Student 15 waits to leave the room
     Student 12 waits to leave the room
     Student 5 enters the room (4 students in the room)
     Student 15 leaves the room (3 students in the room)
     Student 10 enters the room (4 students in the room)
     Student 7 waits to enter the room
     Student 3 waits to leave the room
     Student 3 leaves the room (3 students in the room)
     Student 12 leaves the room (2 students in the room)
     Student 4 enters the room (3 students in the room)
     Student 5 waits to leave the room
     Student 5 leaves the room (2 students in the room)
     Student 15 waits to enter the room
The landlord checks the room the 4 time
The landlord finds there are 2 students in the room (condition being good) and leaves.
     Student 12 waits to enter the room
     Student 4 waits to leave the room
     Student 10 waits to leave the room
     Student 2 enters the room (3 students in the room)
     Student 4 leaves the room (2 students in the room)
     Student 6 enters the room (3 students in the room)
     Student 5 waits to enter the room
     Student 3 waits to enter the room
     Student 9 enters the room (4 students in the room)
     Student 10 leaves the room (3 students in the room)
     Student 2 waits to leave the room
     Student 4 waits to enter the room
     Student 2 leaves the room (2 students in the room)
     Student 6 waits to leave the room
     Student 14 enters the room (3 students in the room)
     Student 9 waits to leave the room
     Student 10 waits to enter the room
     Student 2 waits to enter the room
     Student 11 enters the room (4 students in the room)
     Student 6 leaves the room (3 students in the room)
     Student 14 waits to leave the room
     Student 9 leaves the room (2 students in the room)
     Student 1 enters the room (3 students in the room)
     Student 14 leaves the room (2 students in the room)
     Student 8 enters the room (3 students in the room)
     Student 6 waits to enter the room
     Student 11 waits to leave the room
     Student 9 waits to enter the room
     Student 11 leaves the room (2 students in the room)
     Student 13 enters the room (3 students in the room)
     Student 7 enters the room (4 students in the room)
     Student 14 waits to enter the room
     Student 8 waits to leave the room
     Student 11 waits to enter the room
     Student 8 leaves the room (3 students in the room)
     Student 9 enters the room (4 students in the room)
     Student 1 waits to leave the room
     Student 1 leaves the room (3 students in the room)
     Student 7 waits to leave the room
     Student 8 waits to enter the room
     Student 9 waits to leave the room
     Student 6 enters the room (4 students in the room)
     Student 9 leaves the room (3 students in the room)
```

```
     Student 12 enters the room (4 students in the room)
     Student 13 waits to leave the room
     Student 9 waits to enter the room
     Student 1 waits to enter the room
     Student 6 waits to leave the room
     Student 7 leaves the room (3 students in the room)
     Student 3 enters the room (4 students in the room)
     Student 12 waits to leave the room
     Student 13 leaves the room (3 students in the room)
     Student 5 enters the room (4 students in the room)
     Student 1 enters the room (5 students in the room)
     Student 7 waits to enter the room
     Student 4 enters the room (6 students in the room)
     Student 6 leaves the room (5 students in the room)
     Student 2 enters the room (6 students in the room)
     Student 3 waits to leave the room
     Student 12 leaves the room (5 students in the room)
     Student 3 leaves the room (4 students in the room)
     Student 10 enters the room (5 students in the room)
     Student 15 enters the room (6 students in the room)
     Student 3 waits to enter the room
     Student 5 waits to leave the room
     Student 5 leaves the room (5 students in the room)
     Student 12 waits to enter the room
     Student 13 waits to enter the room
     Student 1 waits to leave the room
     Student 4 waits to leave the room
     Student 6 waits to enter the room
     Student 10 waits to leave the room
     Student 5 waits to enter the room
     Student 1 leaves the room (4 students in the room)
     Student 11 enters the room (5 students in the room)
     Student 4 leaves the room (4 students in the room)
     Student 6 enters the room (5 students in the room)
     Student 2 waits to leave the room
     Student 10 leaves the room (4 students in the room)
     Student 13 enters the room (5 students in the room)
     Student 4 waits to enter the room
     Student 2 leaves the room (4 students in the room)
     Student 1 waits to enter the room
     Student 11 waits to leave the room
     Student 9 enters the room (5 students in the room)
     Student 15 waits to leave the room
     Student 6 waits to leave the room
     Student 11 leaves the room (4 students in the room)
     Student 10 waits to enter the room
The landlord checks the room the 5 time
The landlord finds there are 4 students in the room (condition being good) and leaves.
     Student 13 waits to leave the room
     Student 6 leaves the room (3 students in the room)
     Student 7 enters the room (4 students in the room)
     Student 11 waits to enter the room
     Student 6 terminates
     Student 2 waits to enter the room
     Student 9 waits to leave the room
     Student 7 waits to leave the room
     Student 10 enters the room (5 students in the room)
     Student 13 leaves the room (4 students in the room)
     Student 11 enters the room (5 students in the room)
     Student 15 leaves the room (4 students in the room)
     Student 10 waits to leave the room
     Student 14 enters the room (5 students in the room)
     Student 9 leaves the room (4 students in the room)
     Student 8 enters the room (5 students in the room)
     Student 13 terminates
     Student 15 terminates
     Student 9 terminates
     Student 8 waits to leave the room
     Student 11 waits to leave the room
     Student 5 enters the room (6 students in the room)
     Student 7 leaves the room (5 students in the room)
     Student 4 enters the room (6 students in the room)
     Student 14 waits to leave the room
     Student 1 enters the room (7 students in the room)
     Student 7 terminates
     Student 14 leaves the room (6 students in the room)
     Student 12 enters the room (7 students in the room)
     Student 5 waits to leave the room
     Student 5 leaves the room (6 students in the room)
     Student 8 leaves the room (5 students in the room)
     Student 5 terminates
     Student 12 waits to leave the room
     Student 4 waits to leave the room
     Student 14 terminates
     Student 1 waits to leave the room
     Student 3 enters the room (6 students in the room)
     Student 8 terminates
     Student 2 enters the room (7 students in the room)
     Student 11 leaves the room (6 students in the room)
     Student 11 terminates
     Student 10 leaves the room (5 students in the room)
     Student 3 waits to leave the room
     Student 3 leaves the room (4 students in the room)
     Student 12 leaves the room (3 students in the room)
     Student 3 terminates
     Student 12 terminates
     Student 2 waits to leave the room
     Student 4 leaves the room (2 students in the room)
     Student 10 terminates
     Student 1 leaves the room (1 students in the room)
     Student 2 leaves the room (0 students in the room)
     Student 1 terminates
     Student 4 terminates
     Student 2 terminates
After checking the room 5 times, the landlord retires and is on vacation!
```

✓

```
1. Question: How did you make sure that no students can enter while the landlord is in
the room and checking? Explain your approach in details.

        My approach to solving this problem by not allowing students to enter or leave
while the landlord is actively checking the room is done
        by disabling the progression (by using semaphores) forwards for students. For
the portions of code that dictate when a student enters
        or leaves, I've placed them in Semaphore blocks that will stop the action once
Landlord triggers them. Then throughout the Landlords
        function, I've strategically placed the different types blocking mechanisms that
doesn't allow for the students to progress further until the
        semaphores are released (by the landlord).

2. Question: How did you make sure that the landlord will not leave until all students
have left the room? Explain your approach in details.

        I've placed in a global counting mechanism, using a static variable defined in
the header file, that will add/subtract based at where that
        thread is in code. This global counting system has mechanics in place to make
sure that it will be computed atomically during many thread
        calculations because its placed behind a mutex and semaphore. There is no worry
of a race condition being met, even though this is a
        shared variable.

3. Question: How did you make sure the message "After checking the room XX times, the
landlord retires and is on vacation!" is the last message printed by your program?

        To solve this issue and make sure that the above message will always appear is
the last variable is by placing one global counter that gets
        incremented when the student thread gets spawned (this occurs in main) and it
will get decremented once the student process gets terminated
        (you can find this portion in thread-support, close to the final print
statement). Then inside thread.cpp, when the landlord is finishing up
        it's last iteration, there is a special case inside loop that will handle/wait
for all of the student process to finish up their computations.
        And guarentee that, that message will appear at the vary end.
```

✓

CS3331 Program 4 Grade Report

You receive 0 point if any one of the following occurs
No further grading will be done

Problem	*Check All Apply*	*You Receive*
Not-compile		0
Compile-but-not-run		0
Meaningless and/or vague Program		0
Did not implement the indicated methods		0
Did not follow the required program structure		0
Other significant deviation from specification, e.g., maximum parallelism		0
Totally wrong and unacceptable output		0

This part applies only if you have a working program

Item		*Max Possible*	*You Receuve*
Style & Doc.	Header in each file	2	2
	Good indentation	2	2
	Good comments	2	2
	Good use of function, variable names, etc & no GOTO	2	2
Spec	Handles input and argument list properly	2	2
	Correct output format	5	5
Correctness	Work on sample data	35	35
	Work on our data	35	35
README	Missing README – next two items receive 0	0	0
	Well-written README	5	5
	Answer questions properly	10	10
Deduction	Busy Waiting	-10	
	Race Conditions 10 points each	-10	
	Deadlocks 10 points each	-10	
	Compilation produces visualization	-10	
Total		100	100

Your Score: 100