```
/*
  ============================================================================
  PROJECT: Santa Clause Problem
  ============================================================================
  NAME : Tyler Neal
  User ID : tpneal
  DATE : 04/09/2024
  PROGRAM ASSIGNMENT : #5
  FILE NAME : thread.h
  PROGRAM PURPOSE :
    This program simulates Santa Claus's operations at the North Pole,
    coordinating between Santa, his reindeers, and elves using a Hoare
    type monitor for synchronization. It models Santa being awoken by
    either all reindeers returning from vacation or by elves in groups
    of three with problems, ensuring proper handling of Christmas
    preparations and toy-making issues without deadlock or starvation.
  ============================================================================
*/

#ifndef THREAD_H
#define THREAD_H

#include <ThreadClass.h>
#include <string>
#include <chrono>
#include <thread>
#include <random>
#include <cstdarg>

// The following definition values are arbitrary and may be adjusted as needed
#define MAX_DELAY_SECONDS 5         // Max time delay used for waiting reindeer and elves
#define MAX_ANSWER_QUESTION_TIME 3  // Max time to answer questions
#define MAX_PREP_SLEIGH_TIME 3      // Max time to prepare sleigh
#define MAX_DELIVERY_TIME 5         // Max time to deliver toys

// Monitor class for managing interactions
class MyMonitor : public Monitor {
public:
    MyMonitor(int total_elves, int total_reindeer, int required_deliveries, Mutex* mtx);

    void Sleep();
    void AskQuestion(int id);
    void AnswerQuestion();
    void ReindeerBack(int id);
    void WaitOthers(int id);
    void GatherDeer();
    void WaitSleigh(int id);
    void PrepareSleigh();
    void FlyOff(int id);
    void DeliverToys();

    bool elves_have_question;
    bool all_deer_back;
    bool retired;

private:
    Mutex* print_mutex;
    Condition attention_required;
    Condition santa_ready;
    Condition elf_group_ready;
    Condition reindeer_group_ready;
    Condition sleigh_ready;
    Condition vacation_time;

    int total_elves;
    int total_reindeer;
    int required_deliveries;
    int reindeer_home;
    int question_counter;
    int delivery_count;

    bool santa_sleeping;
    bool santa_home;

    int* elf_queue;
    int elf_queue_iter;
};

// Santa class representing Santa's thread and behaviors
class Santa : public Thread {
public:
    Santa(MyMonitor* monitor, Mutex* mtx);

private:
    void ThreadFunc() override;
    MyMonitor* monitor;
    Mutex* print_mutex;
};

// Elf class representing each elf's thread and behaviors
class Elf : public Thread {
public:
    Elf(int id, MyMonitor* monitor, Mutex* mtx);

private:
    void ThreadFunc() override;
    int id;
    MyMonitor* monitor;
    Mutex* print_mutex;
};

// Reindeer class representing each reindeer's thread and behaviors
class Reindeer : public Thread {
public:
    Reindeer(int id, MyMonitor* monitor, Mutex* mtx);

private:
    void ThreadFunc() override;
    int id;
    MyMonitor* monitor;
    Mutex* print_mutex;
};

// Helper functions for thread operation
void wait(int seconds);
void random_wait(int max_wait);
std::string spaces(int num_spaces);
void printout(int space_count, Mutex* print_mutex, const char* format, ...);

#endif // THREAD_H
```

```
/*
  ========================================================================
  PROJECT: Santa Clause Problem
  ========================================================================
  NAME : Tyler Neal
  User ID : tpneal
  DATE : 04/09/2024
  PROGRAM ASSIGNMENT : #5
  FILE NAME : thread.cpp
  PROGRAM PURPOSE :
    This program simulates Santa Claus's operations at the North Pole,
    coordinating between Santa, his reindeers, and elves using a Hoare
    type monitor for synchronization. It models Santa being awoken by
    either all reindeers returning from vacation or by elves in groups
    of three with problems, ensuring proper handling of Christmas
    preparations and toy-making issues without deadlock or starvation.
  ========================================================================
*/

#include "thread.h"

//========================================================================//
//                              MONITOR                                   //
//========================================================================//
/* Monitor Constructor */
MyMonitor::MyMonitor(int total_elves, int total_reindeer, int required_deliveries, Mutex* mtx)
    : Monitor("monitor", HOARE),

    // Conditions & Mutex
    print_mutex(mtx),
    attention_required("attention_required"),
    santa_ready("santa_ready"),
    elf_group_ready("elf_group_ready"),
    reindeer_group_ready("reindeer_group_ready"),
    sleigh_ready("sleigh_ready"),
    vacation_time("vacation_time"),

    // Integers
    total_elves(total_elves),
    total_reindeer(total_reindeer),
    required_deliveries(required_deliveries),
    question_counter(0),
    reindeer_home(0),
    delivery_count(0),

    // Booleans
    elves_have_question(false),
    all_deer_back(false),
    retired(false),
    santa_sleeping(false),
    santa_home(true),

    // Etc.
    elf_queue(new int[3]),
    elf_queue_iter(0)
{
}

/* Monitor Methods */
void MyMonitor::Sleep()
{
    MonitorBegin();

    // Before sleeping, check if still needed
    if (!elves_have_question && !all_deer_back)
    {
        // Sleep until attention is required
        printout(0, print_mutex, "Santa takes a nap zZz");
        santa_sleeping = true;
        attention_required.Wait();
        printout(0, print_mutex, "Santa wakes up!");
        santa_sleeping = false;
    }

    MonitorEnd();
}

void MyMonitor::AskQuestion(int id)
{
    MonitorBegin();

    // Ask question if santa's home and not already with a group
    if (!elves_have_question && santa_home)
    {
        // Add elf question to queue
        printout(9, print_mutex, "Elf %d has a problem", id);
        question_counter++;
        elf_queue[elf_queue_iter++] = id;

        // Wait until group of three to wake santa
        if (question_counter < 3)
        {
            elf_group_ready.Wait();
        }
        else
        {
            printout(9, print_mutex, "Elves %d, %d, %d, wake up Santa", elf_queue[0], elf_queue[1],
elf_queue[2]);
            elves_have_question = true;
            attention_required.Signal();
        }
    }

    MonitorEnd();
}

void MyMonitor::AnswerQuestion()
{
    MonitorBegin();

    // Santa answers elves questions
    printout(0, print_mutex, "Santa answers the question posted by elves %d, %d, %d", elf_queue[0],
elf_queue[1], elf_queue[2]);
    random_wait(MAX_ANSWER_QUESTION_TIME);     // take time to answer question

    // Reset elf group related variables
    printout(9, print_mutex, "Elves %d, %d, %d, return to work", elf_queue[0], elf_queue[1],
elf_queue[2]);
    elf_queue_iter = 0;
    for (int i = 0; i < question_counter; i++) elf_queue[i] = 0;
    for (int i = 1; i < question_counter; i++) elf_group_ready.Signal();
    question_counter = 0;
    elves_have_question = false;

    MonitorEnd();
}

void MyMonitor::ReindeerBack(int id)
{
    MonitorBegin();

    printout(4, print_mutex, "Reindeer %d returns", id);
    reindeer_home++;

    // If returning reindeer is the last home, wake up santa
    if (reindeer_home >= total_reindeer)
    {
        printout(4, print_mutex, "The last reindeer %d wakes up Santa", id);
        all_deer_back = true;
```

```
        santa_home = false;
        attention_required.Signal();
    }

    MonitorEnd();
}

void MyMonitor::WaitOthers(int id)
{
    MonitorBegin();

    // Wait for all reindeer to return
    if(all_deer_back == false)
    {
        reindeer_group_ready.Wait();
    }

    MonitorEnd();
}

void MyMonitor::GatherDeer()
{
    MonitorBegin();

    // Round up reindeer to wait at the sleigh
    for (int i = 0; i < total_reindeer - 1; i++)
    {
        reindeer_group_ready.Signal();
    }

    MonitorEnd();
}

void MyMonitor::WaitSleigh(int id)
{
    MonitorBegin();

    // Wait for santa to finish preparing the sleigh
    sleigh_ready.Wait();

    MonitorEnd();
}

void MyMonitor::PrepareSleigh()
{
    MonitorBegin();

    // Santa preps the sleigh, and lets reindeer know when he's done
    printout(0, print_mutex, "Santa is preparing the sleigh");
    random_wait(MAX_PREP_SLEIGH_TIME); // take time to prep the sleigh
    for (int i = 0; i < total_reindeer; i++)
    {
        sleigh_ready.Signal();
    }

    MonitorEnd();
}

void MyMonitor::FlyOff(int id)
{
    MonitorBegin();

    // Reindeer take off to deliver toys, and anticipate their vacation
    vacation_time.Wait();

    MonitorEnd();
}

void MyMonitor::DeliverToys()
{
    MonitorBegin();

    // The team takes off to deliver toys
    printout(0, print_mutex, "The team flies off! (%d)", ++delivery_count);
    printout(0, print_mutex, "..........");
    random_wait(MAX_DELIVERY_TIME); // Spend time delivering toys

    // If there's more deliveries to be done reset variables, otherwise retire
    if (delivery_count < required_deliveries)
    {
        all_deer_back = false;
        reindeer_home = 0;
        santa_home = true;
    }
    else
    {
        printout(0, print_mutex, "\nAfter (%d) deliveries, Santa retires and is on vacation!\n",
delivery_count);
        retired = true;
        // Resume waiting elves
        for (int i = 0; i < total_elves; i++)
        {
            elf_group_ready.Signal();
        }
    }

    // Send reindeer off on vacation
    for (int i = 0; i < total_reindeer; i++)
    {
        vacation_time.Signal();
    }

    MonitorEnd();
}

//========================================================================//
//                                SANTA                                   //
//========================================================================//
/* Santa Constructor */
Santa::Santa(MyMonitor* monitor, Mutex* mtx) : monitor(monitor), print_mutex(mtx)
{
    printout(0, print_mutex, "Santa thread starts");
    printout(0, print_mutex, "..........");
}

/* Santa Behavior */
void Santa::ThreadFunc()
{
    Thread::ThreadFunc();

    wait(1); // Allows for clean initial thread creation print (not necessary)

    // Work until all deliveries are completed
    while(!monitor->retired)
    {
        monitor->Sleep();                   // Take a nap
        if (monitor->all_deer_back)         // If deer are back, prep toy delivery
        {
            monitor->GatherDeer();
            monitor->PrepareSleigh();
            monitor->DeliverToys();
        }
        if (monitor->elves_have_question)   // If elves have question, answer them
        {
            monitor->AnswerQuestion();
        }
    }

    Exit();
}
```

```
//========================================================================//
//                                ELF                                     //
//========================================================================//
/* Elf Constructor */
Elf::Elf(int id, MyMonitor* monitor, Mutex* mtx) : id(id), monitor(monitor), print_mutex(mtx)
{
    printout(9, print_mutex, "Elf %d starts", id);
    printout(9, print_mutex, ".........");
}

/* Elf Behavior */
void Elf::ThreadFunc()
{
    Thread::ThreadFunc();

    wait(1); // Allows for clean initial thread creation print (not necessary)

    // Work until santa has retired
    while(1)
    {
        if (monitor->retired) break;
        random_wait(MAX_DELAY_SECONDS);     // Make toys
        if (monitor->retired) break;
        monitor->AskQuestion(id);           // Encountered a problem
        if (monitor->retired) break;
        random_wait(MAX_DELAY_SECONDS);                 // Problem solved, take a rest
    }

    // Terminate
    printout(9, print_mutex, "Elf %d terminates", id);
    Exit();
}


//========================================================================//
//                            REINDEER                                    //
//========================================================================//
/* Reindeer Constructor */
Reindeer::Reindeer(int id, MyMonitor* monitor, Mutex* mtx) : id(id), monitor(monitor),
print_mutex(mtx)
{
    printout(4, print_mutex, "Reindeer %d starts", id);
    printout(4, print_mutex, ".........");
}

/* Reindeer Behavior */
void Reindeer::ThreadFunc()
{
    Thread::ThreadFunc();

    wait(1); // Allows for clean initial thread creation print (not necessary)

    // Work until santa has retired
    while(1)
    {
        if (monitor->retired) break;
        random_wait(MAX_DELAY_SECONDS);     // Tan on the beach
        monitor->ReindeerBack(id);          // Report back to santa
        if(monitor->all_deer_back == false) // If deer still gone, wait for them to arrive
        {
            monitor->WaitOthers(id);
        }
        monitor->WaitSleigh(id);            // Wait for santa to attach sleigh
        monitor->FlyOff(id);                // Go deliver toys, then take a vacation
        if (monitor->retired) break;
        random_wait(MAX_DELAY_SECONDS);     // Prepare for vacation
    }
    // Terminate
    printout(4, print_mutex, "Reindeer %d terminates", id);
    Exit();
}


//========================================================================//
//                              HELPERS                                   //
//========================================================================//
// Wait for a specific number of seconds
void wait(int seconds) {
    std::this_thread::sleep_for(std::chrono::seconds(seconds));
}

// Wait for a random duration up to max_wait seconds
void random_wait(int max_wait) {
    std::random_device rd;
    std::mt19937 gen(rd());
    std::uniform_int_distribution<> distrib(1, max_wait);
    wait(distrib(gen));
}

// Generate a string consisting of a specific number of spaces
std::string spaces(int num_spaces) {
    return std::string(num_spaces, ' ');
}

// Print output with a specific indentation
void printout(int space_count, Mutex* print_mutex, const char* format, ...) {
    const int BUFFER_SIZE = 1024;
    char buffer[BUFFER_SIZE];
    va_list args;
    va_start(args, format);
    vsnprintf(buffer, BUFFER_SIZE, format, args);
    va_end(args);

    print_mutex->Lock();
    std::cout << spaces(space_count) << buffer << std::endl;
    print_mutex->Unlock();
}
```

```
/*
  ============================================================================
  PROJECT: Santa Clause Problem
  ============================================================================
  NAME : Tyler Neal
  User ID : tpneal
  DATE : 04/09/2024
  PROGRAM ASSIGNMENT : #5
  FILE NAME : thread-main.cpp
  PROGRAM PURPOSE :
    This program simulates Santa Claus's operations at the North Pole,
    coordinating between Santa, his reindeers, and elves using a Hoare
    type monitor for synchronization. It models Santa being awoken by
    either all reindeers returning from vacation or by elves in groups
    of three with problems, ensuring proper handling of Christmas
    preparations and toy-making issues without deadlock or starvation.
  ============================================================================
*/

#include "thread.h"

int main (int argc, char* argv[]) {

/* Argument Validity Check */
if (argc < 4 || argc > 4) {
  printf("Usage: ./prog5 num_elves num_reindeer num_toy_deliveries");
  exit(EXIT_FAILURE);
} printf("\n");

/* Parse Arguments */
int num_elves = atoi(argv[1]);
int num_reindeer = atoi(argv[2]);
int num_toy_deliveries = atoi(argv[3]);

if (num_elves == 0) num_elves = 7;
if (num_reindeer == 0) num_reindeer = 9;
if (num_toy_deliveries == 0) num_toy_deliveries = 5;

/* Create Monitor for Thread Managment */
Mutex* print_mutex = new Mutex();
MyMonitor* monitor = new MyMonitor(num_elves, num_reindeer, num_toy_deliveries, print_mutex);

/* Allocate Space for Threads*/
Santa* santa = new Santa(monitor, print_mutex);
Elf** elves = (Elf**) malloc((num_elves) * sizeof(Elf*));
Reindeer** reindeer = (Reindeer**) malloc((num_reindeer) * sizeof(Reindeer*));

/* Create and Start Threads */
santa->Begin();

for (int i = 0; i < num_reindeer; i++)
{
  reindeer[i] = new Reindeer(i+1, monitor, print_mutex);
  reindeer[i]->Begin();
}

for (int i = 0; i < num_elves; i++)
{
  elves[i] = new Elf(i+1, monitor, print_mutex);
  elves[i]->Begin();
}

/* Wait for All Threads to Finish */
santa->Join();

for (int i = 0; i < num_reindeer; i++)
{
  reindeer[i]->Join();
}

for (int i = 0; i < num_elves; i++)
{
  elves[i]->Join();
}

/* Clean Threads and Dynamic Memory */
for (int i = 0; i < num_elves; i++)
{
  if (elves[i])
  {
    delete elves[i];
  }
}

for (int i = 0; i < num_reindeer; i++)
{
  if (reindeer[i])
  {
    delete reindeer[i];
  }
}

delete print_mutex;
delete monitor;
delete santa;
delete[] elves;
delete[] reindeer;

printf("\nExit Success!\n");
return EXIT_SUCCESS;
}
```

```
CC       = c++
FLAGS    = -no-pie
CFLAGS   = -g -O2 -Wno-write-strings -Wno-cpp
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
EXE_FILE = prog5

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

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

thread-main.o: thread-main.cpp thread.h
        ${CC} ${FLAGS} ${DFLAGS} ${IFLAGS} ${CFLAGS} -c thread-main.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 prog5
c++ -no-pie -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 -c
thread.cpp
c++ -no-pie -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 -c
thread-main.cpp
c++ -no-pie  -o prog5  thread.o thread-main.o
/local/eit-linux/apps/ThreadMentor/NoVisual/libthreadclass.a -lpthread
Compilation done.
========================== TEST 1 ==========================
Santa thread starts
..........
    Reindeer 1 starts
    ..........
    Reindeer 2 starts
    ..........
    Reindeer 3 starts
    ..........
    Reindeer 4 starts
    ..........
    Reindeer 5 starts
    ..........
    Reindeer 6 starts
    ..........
    Reindeer 7 starts
    ..........
    Reindeer 8 starts
    ..........
    Reindeer 9 starts
    ..........
         Elf 1 starts
         ..........
         Elf 2 starts
         ..........
         Elf 3 starts
         ..........
         Elf 4 starts
         ..........
         Elf 5 starts
         ..........
         Elf 6 starts
         ..........
         Elf 7 starts
         ..........
Santa takes a nap zZz
         Elf 2 has a problem
    Reindeer 1 returns
    Reindeer 2 returns
    Reindeer 9 returns
         Elf 4 has a problem
    Reindeer 3 returns
    Reindeer 4 returns
    Reindeer 5 returns
    Reindeer 8 returns
         Elf 6 has a problem
         Elves 2, 4, 6, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 2, 4, 6
         Elves 2, 4, 6, return to work
    Reindeer 7 returns
    Reindeer 6 returns
    The last reindeer 6 wakes up Santa
Santa is preparing the sleigh
The team flies off! (1)
..........
Santa takes a nap zZz
         Elf 7 has a problem
    Reindeer 8 returns
         Elf 5 has a problem
         Elf 6 has a problem
         Elves 7, 5, 6, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 7, 5, 6
         Elves 7, 5, 6, return to work
    Reindeer 3 returns
Santa takes a nap zZz
    Reindeer 7 returns
         Elf 4 has a problem
         Elf 2 has a problem
    Reindeer 6 returns
    Reindeer 1 returns
    Reindeer 2 returns
    Reindeer 9 returns
    Reindeer 5 returns
         Elf 3 has a problem
         Elves 4, 2, 3, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 4, 2, 3
         Elves 4, 2, 3, return to work
    Reindeer 4 returns
    The last reindeer 4 wakes up Santa
Santa is preparing the sleigh
The team flies off! (2)
..........
         Elf 4 has a problem
Santa takes a nap zZz
         Elf 5 has a problem
         Elf 3 has a problem
         Elves 4, 5, 3, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 4, 5, 3
         Elves 4, 5, 3, return to work
    Reindeer 8 returns
    Reindeer 7 returns
Santa takes a nap zZz
         Elf 7 has a problem
    Reindeer 9 returns
    Reindeer 4 returns
         Elf 6 has a problem
    Reindeer 2 returns
    Reindeer 5 returns
    Reindeer 3 returns
         Elf 4 has a problem
         Elves 7, 6, 4, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 7, 6, 4
         Elves 7, 6, 4, return to work
    Reindeer 6 returns
    Reindeer 1 returns
    The last reindeer 1 wakes up Santa
Santa is preparing the sleigh
The team flies off! (3)
..........
         Elf 7 has a problem
Santa takes a nap zZz
         Elf 2 has a problem
         Elf 3 has a problem
         Elves 7, 2, 3, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 7, 2, 3
         Elves 7, 2, 3, return to work
    Reindeer 3 returns
    Reindeer 7 returns
Santa takes a nap zZz
         Elf 4 has a problem
         Elf 5 has a problem
```

```
    Reindeer 4 returns
         Elf 6 has a problem
         Elves 4, 5, 6, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 4, 5, 6
         Elves 4, 5, 6, return to work
    Reindeer 8 returns
    Reindeer 1 returns
    Reindeer 5 returns
    Reindeer 6 returns
    Reindeer 9 returns
    Reindeer 2 returns
    The last reindeer 2 wakes up Santa
Santa is preparing the sleigh
The team flies off! (4)
..........
         Elf 2 has a problem
         Elf 3 has a problem
         Elf 1 has a problem
         Elves 2, 3, 1, wake up Santa
Santa answers the question posted by elves 2, 3, 1
         Elves 2, 3, 1, return to work
    Reindeer 3 returns
    Reindeer 4 returns
Santa takes a nap zZz
    Reindeer 2 returns
    Reindeer 8 returns
         Elf 7 has a problem
    Reindeer 9 returns
    Reindeer 7 returns
    Reindeer 1 returns
    Reindeer 6 returns
         Elf 6 has a problem
    Reindeer 5 returns
    The last reindeer 5 wakes up Santa
Santa wakes up!
Santa is preparing the sleigh
The team flies off! (5)
..........
         Elf 1 has a problem
         Elves 7, 6, 1, wake up Santa
Santa answers the question posted by elves 7, 6, 1
         Elves 7, 6, 1, return to work
    Reindeer 3 returns
    Reindeer 1 returns
Santa takes a nap zZz
         Elf 1 has a problem
         Elf 2 has a problem
         Elf 4 has a problem
         Elves 1, 2, 4, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 1, 2, 4
         Elves 1, 2, 4, return to work
    Reindeer 7 returns
    Reindeer 4 returns
    Reindeer 2 returns
Santa takes a nap zZz
         Elf 5 has a problem
         Elf 7 has a problem
         Elf 6 has a problem
         Elves 5, 7, 6, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 5, 7, 6
         Elves 5, 7, 6, return to work
    Reindeer 9 returns
Santa takes a nap zZz
    Reindeer 5 returns
    Reindeer 8 returns
    Reindeer 6 returns
    The last reindeer 6 wakes up Santa
Santa wakes up!
Santa is preparing the sleigh
The team flies off! (6)
..........
         Elf 1 has a problem
         Elf 2 has a problem
         Elf 7 has a problem
         Elves 1, 2, 7, wake up Santa
Santa answers the question posted by elves 1, 2, 7
         Elves 1, 2, 7, return to work
    Reindeer 1 returns
    Reindeer 3 returns
    Reindeer 2 returns
Santa takes a nap zZz
    Reindeer 9 returns
    Reindeer 5 returns
    Reindeer 8 returns
         Elf 3 has a problem
         Elf 7 has a problem
    Reindeer 7 returns
    Reindeer 4 returns
         Elf 1 has a problem
         Elves 3, 7, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 7, 1
         Elves 3, 7, 1, return to work
Santa takes a nap zZz
    Reindeer 6 returns
    The last reindeer 6 wakes up Santa
Santa wakes up!
Santa is preparing the sleigh
The team flies off! (7)
..........
Santa takes a nap zZz
         Elf 3 has a problem
         Elf 5 has a problem
    Reindeer 6 returns
    Reindeer 9 returns
    Reindeer 7 returns
         Elf 2 has a problem
         Elves 3, 5, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 5, 2
         Elves 3, 5, 2, return to work
    Reindeer 4 returns
    Reindeer 1 returns
    Reindeer 8 returns
Santa takes a nap zZz
    Reindeer 3 returns
    Reindeer 2 returns
         Elf 6 has a problem
    Reindeer 5 returns
    The last reindeer 5 wakes up Santa
Santa wakes up!
Santa is preparing the sleigh
The team flies off! (8)
..........

After (8) deliveries, Santa retires and is on vacation!

         Elf 6 terminates
         Elf 7 terminates
    Reindeer 6 terminates
    Reindeer 9 terminates
    Reindeer 5 terminates
    Reindeer 7 terminates
    Reindeer 4 terminates
    Reindeer 1 terminates
    Reindeer 8 terminates
    Reindeer 3 terminates
```

last msg -2

```
    Reindeer 2 terminates
         Elf 4 terminates
         Elf 3 terminates
         Elf 1 terminates
         Elf 5 terminates
         Elf 2 terminates

Exit Success!

============================ TEST 2 ============================

Santa thread starts
..........
    Reindeer 1 starts
    ..........
    Reindeer 2 starts
    ..........
    Reindeer 3 starts
    ..........
    Reindeer 4 starts
    ..........
    Reindeer 5 starts
    ..........
         Elf 1 starts
         ..........
         Elf 2 starts
         ..........
         Elf 3 starts
         ..........
Santa takes a nap zZz
         Elf 2 has a problem
         Elf 3 has a problem
         Elf 1 has a problem
         Elves 2, 3, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 2, 3, 1
         Elves 2, 3, 1, return to work
    Reindeer 2 returns
    Reindeer 3 returns
    Reindeer 4 returns
    Reindeer 1 returns
    Reindeer 5 returns
    The last reindeer 5 wakes up Santa
Santa is preparing the sleigh
The team flies off! (1)
..........
Santa takes a nap zZz
         Elf 3 has a problem
    Reindeer 2 returns
         Elf 2 has a problem
    Reindeer 5 returns
    Reindeer 3 returns
         Elf 1 has a problem
         Elves 3, 2, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 2, 1
         Elves 3, 2, 1, return to work
    Reindeer 4 returns
Santa takes a nap zZz
    Reindeer 1 returns
    The last reindeer 1 wakes up Santa
Santa wakes up!
Santa is preparing the sleigh
The team flies off! (2)
..........
         Elf 1 has a problem
         Elf 2 has a problem
Santa takes a nap zZz
         Elf 3 has a problem
         Elves 1, 2, 3, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 1, 2, 3
         Elves 1, 2, 3, return to work
    Reindeer 2 returns
Santa takes a nap zZz
    Reindeer 3 returns
    Reindeer 5 returns
    Reindeer 4 returns
         Elf 3 has a problem
         Elf 2 has a problem
         Elf 1 has a problem
         Elves 3, 2, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 2, 1
         Elves 3, 2, 1, return to work
Santa takes a nap zZz
         Elf 2 has a problem
         Elf 3 has a problem
         Elf 1 has a problem
         Elves 2, 3, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 2, 3, 1
         Elves 2, 3, 1, return to work
Santa takes a nap zZz
         Elf 2 has a problem
         Elf 1 has a problem
         Elf 3 has a problem
         Elves 2, 1, 3, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 2, 1, 3
         Elves 2, 1, 3, return to work
Santa takes a nap zZz
         Elf 1 has a problem
         Elf 3 has a problem
         Elf 2 has a problem
         Elves 1, 3, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 1, 3, 2
         Elves 1, 3, 2, return to work
Santa takes a nap zZz
         Elf 1 has a problem
         Elf 3 has a problem
         Elf 2 has a problem
         Elves 1, 3, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 1, 3, 2
         Elves 1, 3, 2, return to work
Santa takes a nap zZz
         Elf 3 has a problem
         Elf 2 has a problem
         Elf 1 has a problem
         Elves 3, 2, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 2, 1
         Elves 3, 2, 1, return to work
Santa takes a nap zZz
         Elf 3 has a problem
         Elf 2 has a problem
         Elf 1 has a problem
         Elves 3, 2, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 2, 1
         Elves 3, 2, 1, return to work
Santa takes a nap zZz
         Elf 2 has a problem
         Elf 3 has a problem
         Elf 1 has a problem
         Elves 2, 3, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 2, 3, 1
```

```
        Elves 2, 3, 1, return to work
Santa takes a nap zZz
        Elf 2 has a problem
        Elf 3 has a problem
        Elf 1 has a problem
        Elves 2, 3, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 2, 3, 1
        Elves 2, 3, 1, return to work
Santa takes a nap zZz
        Elf 2 has a problem
        Elf 3 has a problem
        Elf 1 has a problem
        Elves 2, 3, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 2, 3, 1
        Elves 2, 3, 1, return to work
Santa takes a nap zZz
        Elf 2 has a problem
        Elf 1 has a problem
        Elf 3 has a problem
        Elves 2, 1, 3, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 2, 1, 3
        Elves 2, 1, 3, return to work
Santa takes a nap zZz
        Elf 2 has a problem
        Elf 3 has a problem
        Elf 1 has a problem
        Elves 2, 3, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 2, 3, 1
        Elves 2, 3, 1, return to work
Santa takes a nap zZz
        Elf 2 has a problem
        Elf 1 has a problem
        Elf 3 has a problem
        Elves 2, 1, 3, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 2, 1, 3
        Elves 2, 1, 3, return to work
Santa takes a nap zZz
        Elf 1 has a problem
        Elf 2 has a problem
        Elf 3 has a problem
        Elves 1, 2, 3, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 1, 2, 3
        Elves 1, 2, 3, return to work
Santa takes a nap zZz
        Elf 2 has a problem
        Elf 1 has a problem
        Elf 3 has a problem
        Elves 2, 1, 3, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 2, 1, 3
        Elves 2, 1, 3, return to work
Santa takes a nap zZz
        Elf 1 has a problem
        Elf 3 has a problem
        Elf 2 has a problem
        Elves 1, 3, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 1, 3, 2
        Elves 1, 3, 2, return to work
Santa takes a nap zZz
        Elf 1 has a problem
        Elf 3 has a problem
        Elf 2 has a problem
        Elves 1, 3, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 1, 3, 2
        Elves 1, 3, 2, return to work
Santa takes a nap zZz
        Elf 1 has a problem
        Elf 3 has a problem
        Elf 2 has a problem
        Elves 1, 3, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 1, 3, 2
        Elves 1, 3, 2, return to work
Santa takes a nap zZz
        Elf 3 has a problem
        Elf 1 has a problem
        Elf 2 has a problem
        Elves 3, 1, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 1, 2
        Elves 3, 1, 2, return to work
Santa takes a nap zZz
        Elf 3 has a problem
        Elf 2 has a problem
        Elf 1 has a problem
        Elves 3, 2, 1, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 2, 1
        Elves 3, 2, 1, return to work
Santa takes a nap zZz
        Elf 3 has a problem
        Elf 1 has a problem
        Elf 2 has a problem
        Elves 3, 1, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 1, 2
        Elves 3, 1, 2, return to work
Santa takes a nap zZz
        Elf 3 has a problem
        Elf 1 has a problem
        Elf 2 has a problem
        Elves 3, 1, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 1, 2
        Elves 3, 1, 2, return to work
Santa takes a nap zZz
        Elf 1 has a problem
        Elf 3 has a problem
        Elf 2 has a problem
        Elves 1, 3, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 1, 3, 2
        Elves 1, 3, 2, return to work
Santa takes a nap zZz
        Elf 1 has a problem
        Elf 2 has a problem
        Elf 3 has a problem
        Elves 1, 2, 3, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 1, 2, 3
        Elves 1, 2, 3, return to work
Santa takes a nap zZz
        Elf 3 has a problem
        Elf 1 has a problem
        Elf 2 has a problem
        Elves 3, 1, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 1, 2
        Elves 3, 1, 2, return to work
Santa takes a nap zZz
        Elf 3 has a problem
        Elf 1 has a problem
```

```
        Elf 2 has a problem
        Elves 3, 1, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 1, 2
        Elves 3, 1, 2, return to work
Santa takes a nap zZz
        Elf 3 has a problem
        Elf 1 has a problem
        Elf 2 has a problem
        Elves 3, 1, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 1, 2
        Elves 3, 1, 2, return to work
Santa takes a nap zZz
        Elf 1 has a problem
        Elf 2 has a problem
        Elf 3 has a problem
        Elves 1, 2, 3, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 1, 2, 3
        Elves 1, 2, 3, return to work
Santa takes a nap zZz
        Elf 3 has a problem
        Elf 1 has a problem
        Elf 2 has a problem
        Elves 3, 1, 2, wake up Santa
Santa wakes up!
Santa answers the question posted by elves 3, 1, 2
        Elves 3, 1, 2, return to work
Santa takes a nap zZz
        Elf 2 has a problem
```

endless looping -10

1. Question:
How do you guarantee that only three elves will ask questions?

Answer: In order to make sure that only three elves will ask a question at a time I keep track of a counter inside of the AskQuestion() function. Each time that an elf asks a question, the counter is incremented, and then the elf waits to be assisted. Once the third elf encounters a question, it signals to santa for help. Santa then answers the question, and resets the counters in order to allow the the next set of elves to answer the question. I also keep a boolean which is toggled off when a group of elves ask a question, and toggled back on by santa when he is finished helping the group of elves. This prevents elves from trying to enter the question queue before it has been reset.

2. Question:
Show how no elf will leave before the questions are answered.

Answer: In order to prevent elves from leaving before their quiestion is answered, they simply wait on a Condition variable. They begin waiting on this condition when they ask a question, and are only signaled/released when santa has finished helping them. The third elf in a set to ask a question does not wait upon this condition, as they are in charge of signaling santa clause. Instead, by signaling santa clause, it's ensured that this elf doesn't resume until santa clause has finished his AnswerQuestion() monitor function.

3. Question:
Show that while three elves are waiting for an answer, no other elves can cut in and ask questions.

Answer: To do this, as stated earlier, I use a boolean value. This boolean value true by default, and encapsulates the entirety of the AskQuestion monitor function. When a group of three elves have a question, the third elf toggles this boolean to false before signaling Santa. Santa then flips this value to true again after he has finished helping answer the elves' question.

4. Question:
How do you guarantee that Santa only answers question while he is not sleeping.

Answer: In order to make sure that santa only answers questions when hes awake, I make sure that the last elf signals santa to wake up from his sleep. This is done through a conditional that santa waits on while he's "sleeping." There is also another condition that must evaluate true in order to ask a question. This condition essentially evaluates if santa is currently sleeping, and isn't out on a toy delivery mission.

5. Question:
Show that when Santa is wakened up by a reindeer, this reindeer is the last one coming back from vacation.

Answer: Similar to how I ensure the third elf is the one to signal santa, I also use this strategy to ensure that the last reindeer is the one to signal Santa. I count the number of reindeer who have returned from vacation. After they have returned, they wait on the remaining reindeer to return. The last reindeer signals Santa through the use of a condition letting him know that all of the reindeer are back and ready to get going.

6. Question:
How do you make sure that Santa always handles reindeers first.

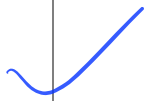
Answer: This is pretty simple. Santa is in a constant loop until retirement. First, he takes a nap. When the conditional is signaled that indicates his attention is required, he needs to figure out what requires his attention. By placing the if statment evaluating if the reindeer are back directly after Santa's call to sleep, he first is able to evaluate if the reindeer are ready to take off. Whether this evaluates true or false does not matter, as afterwards he checks the if statments indicating if the elves need help. Once these are both handled, Santa loops, once again taking a nap.

7. Question:
Show that while Santa is attaching the sleigh and delivering toys, all reindeers are there. They won't sneak out for vacation.

Answer: Once all of the reindeer are back, they all wait on the same conditionals at the same time. There is a conditional that the reindeer wait on while the sleigh is being prepared by Santa. When santa is done with this he signals the reindeer. After they reindeer are active, they immediatly wait on another condition. They wait on this condition while their on the job, and once finished, Santa signals all the reindeer letting them know they did a good job, and that they can go on another vacation. During this whole period the reindeer all act as a group, and proceed in the same way.

8. Question:
Show that while Santa is attaching the sleigh and delivering toys, elves will not ask questions.

Answer: I complete this through the simple use of a boolean. This boolean indicates wether or not santa is home and available to ask questions. When the last reindeer signals santa that all the deer are back, the deer toggles this boolean to let the elves know that they must wait until after the toys are delivered in order to ask him another question.



CS3331 Program 5 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 (e.g., non-Hoare type monitor)		0
Used semaphores other than locking `stdout`		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 Receive*
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	33
	Work on our data	35	25
`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	
Total		100	88

Your Score: 88