Cameron Meyer
CS 4348.001

Project #2 Design

Semaphores:

- `awaitReceptionist`
  - *Purpose:* Indicates that a patient would like to speak with the receptionist. The receptionist waits on this semaphore before retrieving a patient from the queue.
  - *Initial Value:* 0
- `registerPatient`
  - *Purpose:* Indicates that a patient has successfully been registered by the receptionist. The patient waits on this semaphore before sitting in the waiting room.
  - *Initial Value:* 0 (for all elements)
- `receptionDeskDeparture`
  - *Purpose:* Indicates that a patient has finished speaking to the receptionist and has left the desk to sit in the waiting room. The receptionist waits on this semaphore before signaling a nurse that a patient is ready to be seen.
  - *Initial Value:* 0 (for all elements)
- `awaitNurse`
  - *Purpose:* Indicates that a patient is in the waiting room and is ready to see their doctor. The nurse waits on this semaphore before walking a patient to their doctor's office.
  - *Initial Value:* 0 (for all elements)
- `waitingRoomDeparture`
  - *Purpose:* Indicates that a nurse has begun walking a patient to the doctor's office. The patient waits on this semaphore before entering the doctor's office.
  - *Initial Value:* 0 (for all elements)
- `doctorOfficeArrival`
  - *Purpose:* Indicates that a patient has entered the doctor's office. The nurse waits on this semaphore before signaling the doctor to begin the appointment.
  - *Initial Value:* 0 (for all elements)
- `doctorBeginAppointment`
  - *Purpose:* Indicates that a doctor may begin the appointment with the patient in their office. The doctor waits on this semaphore before listening to a patient's symptoms.
  - *Initial Value:* 0 (for all elements)
- `listenSymptoms`
  - *Purpose:* Indicates that a doctor is listening to a patient's symptoms. The patient waits on this semaphore before receiving advice from the doctor.
  - *Initial Value:* 0 (for all elements)
- `listenAdvice`

    - *Purpose:* Indicates that a patient is listening to a doctor's advice. The doctor waits on this semaphore before concluding the appointment.
    - *Initial Value:* 0 (for all elements)
- `patientDeparture`
    - *Purpose:* Indicates that a patient has left the clinic. The doctor waits on this semaphore before ending the appointment.
    - *Initial Value:* 0 (for all elements)
- `doctorEndAppointment`
    - *Purpose:* Indicates that a doctor's appointment has concluded. The nurse waits on this semaphore before checking for new patients.
    - *Initial Value:* 0 (for all elements)
- `adjustDesiredDoctors`
    - *Purpose:* Allows mutually exclusive access to the desiredDoctors array, which is updated by each patient to indicate what doctor they intend to meet with. The receptionist will access this array to determine which nurse to notify that a patient is ready to be walked to the doctor's office.
    - *Initial Value:* 1
- `adjustAwaitingRegistration`
    - *Purpose:* Allows mutually exclusive access to the patientsAwaitingRegistration queue, which keeps track of the order that patients entered the clinic and are waiting to be seen by the receptionist. The receptionist will access and update this queue as they meet with each patient.
    - *Initial Value:* 1
- `adjustAwaitingDoctor`
    - *Purpose:* Allows mutually exclusive access to the patientsAwaitingDoctor queue, which keeps track of the order that patients will meet with a particular doctor (and nurse). The nurse will access and update this queue as they walk each patient to their doctor's office.
    - *Initial Value:* 1
- `adjustInAppointment`
    - *Purpose:* Allows mutually exclusive access to the patientsInAppointment array, which is updated by the nurse to indicate which doctors are currently meeting with a patient. The doctor will access this array to determine which patient they are meeting with.
    - *Initial Value:* 1

Pseudocode:

```
void patient(patientID)
{
      assignDoctor();
      semWait(adjustDesiredDoctors);
      adjustDesiredDoctors();
      semSignal(adjustDesiredDoctors);
      semSignal(awaitReceptionist);
```

```
	semWait(registerPatient[patientID]);
	semSignal(receptionDeskDeparture[patientID]);
	semWait(waitingRoomDeparture[doctorID]);
	semSignal(doctorOfficeArrival[doctorID]);
	semWait(listenSymptoms[doctorID]);
	semSignal(listenAdvice[doctorID]);
	semSignal(patientDeparture[doctorID]);
}

void receptionist()
{
	while(true)
	{
		semWait(awaitReceptionist);
		semWait(adjustAwaitingRegistration);
		selectWaitingPatient();
		semSignal(adjustAwaitingRegistration);
		semSignal(registerPatient[patientID]);
		semWait(receptionDeskDeparture[patientID]);
		semWait(adjustAwaitingDoctor);
		patientEnqueuedForDoctor();
		semSignal(adjustAwaitingDoctor);
		semSignal(awaitNurse[doctorID]);
	}
}

void nurse(nurseID)
{
	while(true)
	{
		semWait(awaitNurse[nurseID]);
		semWait(adjustAwaitingDoctor);
		patientDequeuedForDoctor();
		semSignal(adjustAwaitingDoctor);
		semSignal(waitingRoomDeparture[nurseID]);
		semWait(doctorOfficeArrival[nurseID]);
		semWait(adjustInAppointment);
		patientReadyForAppointment();
		semSignal(adjustInAppointment);
		semSignal(doctorBeginAppointment[nurseID]);
		semWait(doctorEndAppointment[nurseID]);
	}
}
```

```
void doctor(doctorID)
{
	while(true)
	{
		semWait(doctorBeginAppointment[doctorID]);
		semWait(adjustInAppointment);
		retrievePatientInOffice();
		semSignal(adjustInAppointment);
		semSignal(listenSymptoms[doctorID]);
		semWait(listenAdvice[doctorID]);
		semWait(patientDeparture[doctorID]);
		semSignal(doctorEndAppointment[doctorID]);
	}
}
```