**SEMAPHORES:**

**customerSemaphore:**

Purpose: Limits the number of customers that can be inside the post office at a time.

Initial value: 10

**postalWorkerSemaphore:**

Purpose: Controls the availability of postal workers to serve customers. Each postal worker acquires this semaphore before starting a task and releases it after finishing the task.

Initial value: Number of postal workers (3 in this case)

**taskAvailableSemaphore:**

Purpose: Used to signal the availability of tasks for postal workers. Each customer increments this semaphore after submitting their task, and each postal worker decrements this semaphore when taking a task from the queue.

Initial value: 0

**scalesAccessSemaphore:**

Purpose: Controls access to the scales resource. It ensures that only one postal worker can use the scales at a time when mailing a package.

Initial value: 1

**PSUEDOCODE:**

initialize customerSemaphore(10)

initialize postalWorkerSemaphore(3)

initialize taskAvailableSemaphore(0)

initialize scalesAccessSemaphore(1)

initialize taskQueue

create and start postalWorkerThreads[3]

for i = 0 to 49:

create and start customerThread[i]

sleep(1000 / 60)

wait for all customerThread[] to finish

terminate all postalWorkerThreads[]

print "Finished serving customers! Ending simulation.."

customerThread:

print "Customer i created"

wait(customerSemaphore)

print "Customer i enters post office"

enqueue task to taskQueue

signal(taskAvailableSemaphore)

wait(postalWorkerSemaphore)

print "Customer i leaves post office"

signal(customerSemaphore)

postalWorkerThread:

print "Postal worker i created"

while not interrupted:

wait(taskAvailableSemaphore)

task = dequeue from taskQueue

wait(postalWorkerSemaphore)

print "Postal worker i serving customer"

switch(task):

case "buy stamps":

perform buy stamps task

case "mail a letter":

perform mail a letter task

case "mail a package":

wait(scalesAccessSemaphore)

perform mail a package task

signal(scalesAccessSemaphore)

print "Postal worker i finished serving customer"

signal(postalWorkerSemaphore)