Design

/\*program Project\*/

Semaphore persons = 0 //persons will signal once they leave elevator to their floor

Semaphore elevator = 0 // persons will use once they are in elevator for elevator to continue

Semaphore accessElevator = 0 //elevator will grant access to persons by signaling 7 times

Semaphore terminate = 0 //elevator signals to main to terminate program once complete

Semaphore[] sem = Semaphore[9] //array of 9 semaphores, one semaphore for each floor 2-10

FOR (i=0; i<9; i++)

Sem[i] = (new Semaphore=0)

END FOR

ArrayList<Integer> numbers //list that holds person’s floors that are granted access to elevator

HashSet<Integer> hash //similar to numbers but without duplicates

**Class Person extends Thread**(int iD){

int iD

int floor //assign a random floor

Boolean notRide=true

WHILE(notRide)

accesselevator.wait()

add floor to numbers list

print(Customer this.iD entered elevator to go to floor this.floor)

elevator.signal()

sem[this.floor-2].wait() //waits until elevator sends the floor’s semaphore signal

print(Customer this.iD leaves elevator)

this.notRide=false

persons.release() //signals person has left elevator

END WHILE

}

**Class Elevator extends Thread()**{

WHILE (there are people)

accessElevator.signal() 7 times //grants 7 people access to elevator

elevator.wait 7 times //wait for people to get on

sort the numbers list

add integers from numbers list to hash list //gets rid of duplicates

FOR(every integer p in hash set)

print(Elevator door opens at floor ‘p’)

int num = how many times p appears in numbers list

for(num times)

sem[p-2].signal() //floor semaphore signals people to get off

for(num times)

persons.wait() //wait for people to get off

print(Elevator door closes)

clear numbers list

clear hash set

END WHILE

terminate.signal()

}

**void main()**{

create Project project

run project

terminate.wait()

print(Simulation ended)

system.exit(0)

}

**void run()**{

create Elevator elevator thread

start elevator thread

create 49 persons thread

start each person thread

}