Project #2

Multicore Computing

Problem 1

Date	May 17, 2020
Instructor	Bongsoo Sohn
Std. Name	Junhyuck Woo
Std. ID	20145337

INDEX

ENVIRONMENT

Hardware

- MacBook Pro (15-inch, 2017)
- Processor: 2.8 GHz Quad-Core Intel Core i7
- Memory: 16GB 2133 MHz LPDDR3

Operating System

- macOS Catalina, ver: 10.15.4

IDE (Integrated Development Environment)

- IntelliJ IDEA 2019.3.4 (Ultimate Edition)
- Java version “11.0.6”

Testing Environment

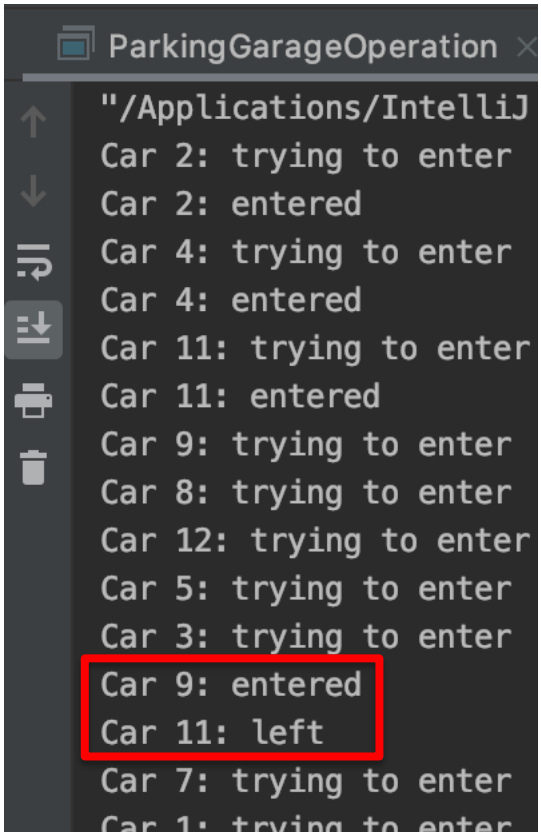
- iTerm2
- Build 3.3.9
- openjdk 14.0.1 2020-04-14
  OpenJDK Runtime Environment (build 14.0.1+7)
  OpenJDK 64-Bit Server VM (build 14.0.1+7, mixed mode, sharing)

EXPLANATION OF RESULT

1. Printing Issue

Before starting the project2, I studied ParkingGarageOperation.java. The guideline required that the number of cars is 40 and the number of parking places is 10. However, it is tough to follow to understand the result, so I reduced the size to the number of cars is 12, and the number of parking places is 3 for easily understand.

I could find the out-of-order result; please check the red box. Car 11 should leave the garage first, and after that, Car 9 could enter. Because there are only 3 places to park, and Car 2, 4, and 11 were already parked. As followed a result, I could find that it is hard to use System.out.println function as we want, and I assumed that it is not a core issue of the assignment.



▲ Result of "ParkingGarageOperation.java" with reduced scale of object

2. Ambiguous Guideline

In the guideline, there are two requirements. The first one is that it requests to refer the two examples: ArrayBlockingQueueExample.java and ParkingGarageOperation.java. The second one is that it required not to use the wait()/notify instead of using java.util.concurrent.

These simple requirements bring ambiguously. It could mean that it uses a producer-consumer structure with BlockingQueue or use the BlockingQueue without the producer-consumer structure.

For these two reasons, I implemented the two software.
ParkingGarageSimulation.java is a software that is just using blockingqueue.
ParkingGarageSimulationPC.java is a software which is using blokcingqueue with producer-consumer structure.

SOURCE CODE

1. ParkingGarageSimulationPC.java

```
// Writer: Junhyuck Woo
// Lecture: Multicore Computing
// Organization: Chung-Ang University
// Deadline: May 17, 2020
// Project #2
// - BlockingQueue with Producer-Consumer Structure

import java.util.concurrent.ArrayBlockingQueue;
import java.util.concurrent.BlockingQueue;

public class ParkingGarageSimulationPC {
    public static void main(String[] args) {
        // Create hte blocking queue
        BlockingQueue places = new ArrayBlockingQueue<String>(10);

        // Create the garage - consumer
        Garage garage = new Garage(places);

        // Create the Car - producer
        for (int i=1; i<=40; i++){
            Car c = new Car("Car " +i, places);
        }
    }
}

class Garage extends Thread{

    // Variable
    private BlockingQueue places;

    // Constructor
    Garage(BlockingQueue places) {
        this.places = places;
        start();
    }

    public void leave() {
        synchronized (System.out) {
            try {
                // Print the status of car: LEAVE
                System.out.println(this.places.take() + ": left");
            }
            catch (InterruptedException e) {}
        }
    }

    public void run() { // enter parking garage
        while (true) {
            try {
                if (this.places.size() != 0) {
                    sleep((int)Math.random()*10000);
                    leave();
                }
            }
            catch (InterruptedException e) {}
        }
    }
}
```

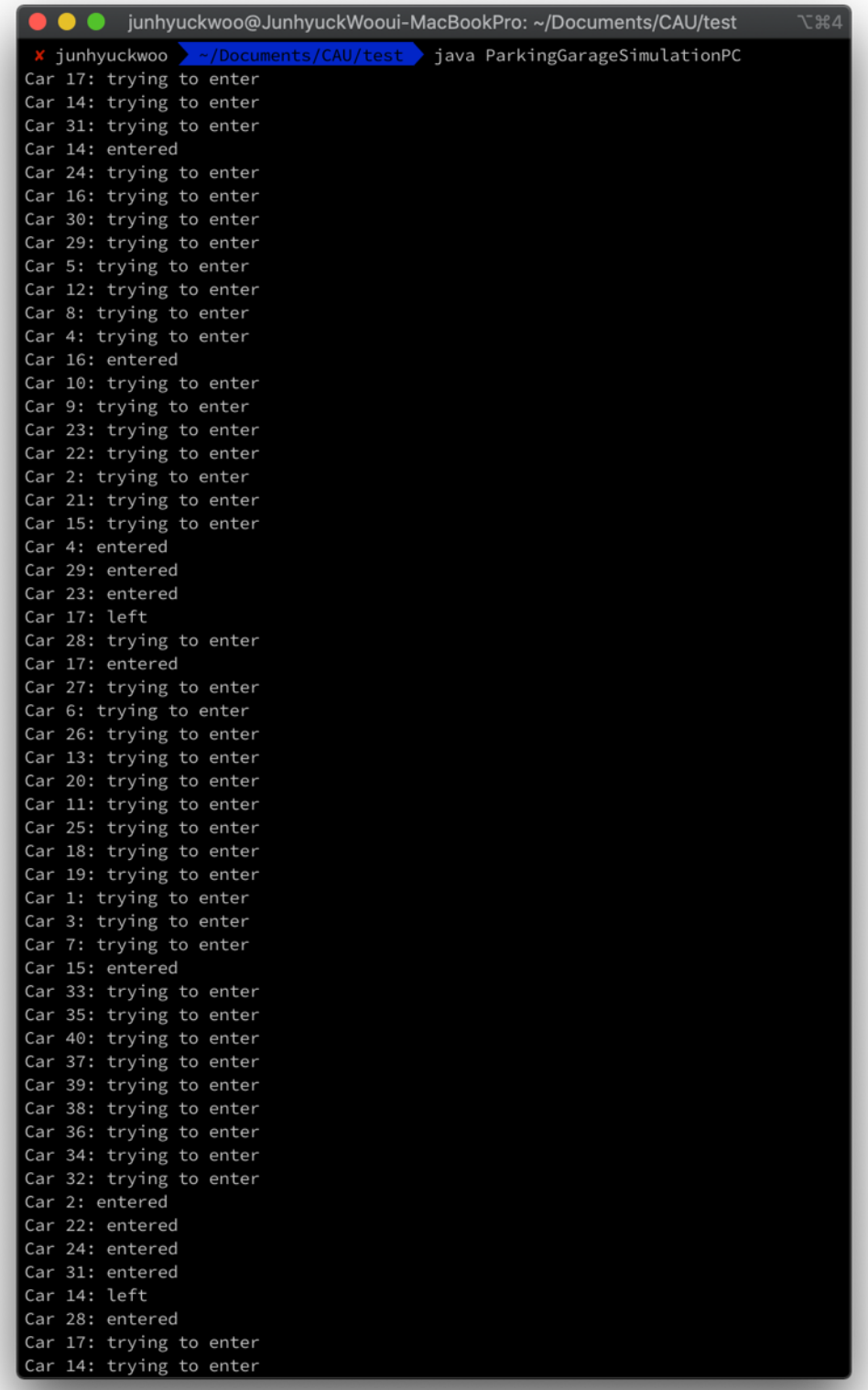
```
class Car extends Thread {
    // Variable
    private BlockingQueue places;

    // Constructor
    public Car(String name, BlockingQueue places) {
        super(name);
        this.places = places;
        start();
    }

    public synchronized void enter() {
        try{
            this.places.put(getName());
            synchronized (System.out) {
                // Print the status of car: ENTER
                System.out.println(getName() + ": entered");
            }
        }
        catch (InterruptedException e) {}
    }

    // Runner
    public void run() {
        while (true) {
            try {
                if (!this.places.contains(getName())){
                    sleep((int)Math.random()*10000);
                    // Print the status of car: TRY TO ENTER
                    System.out.println(getName() + ": trying to enter");
                    enter();
                }
            }
            catch (InterruptedException e) {}
        }
    }
}
```

2. ParkingGarageSimulation.java

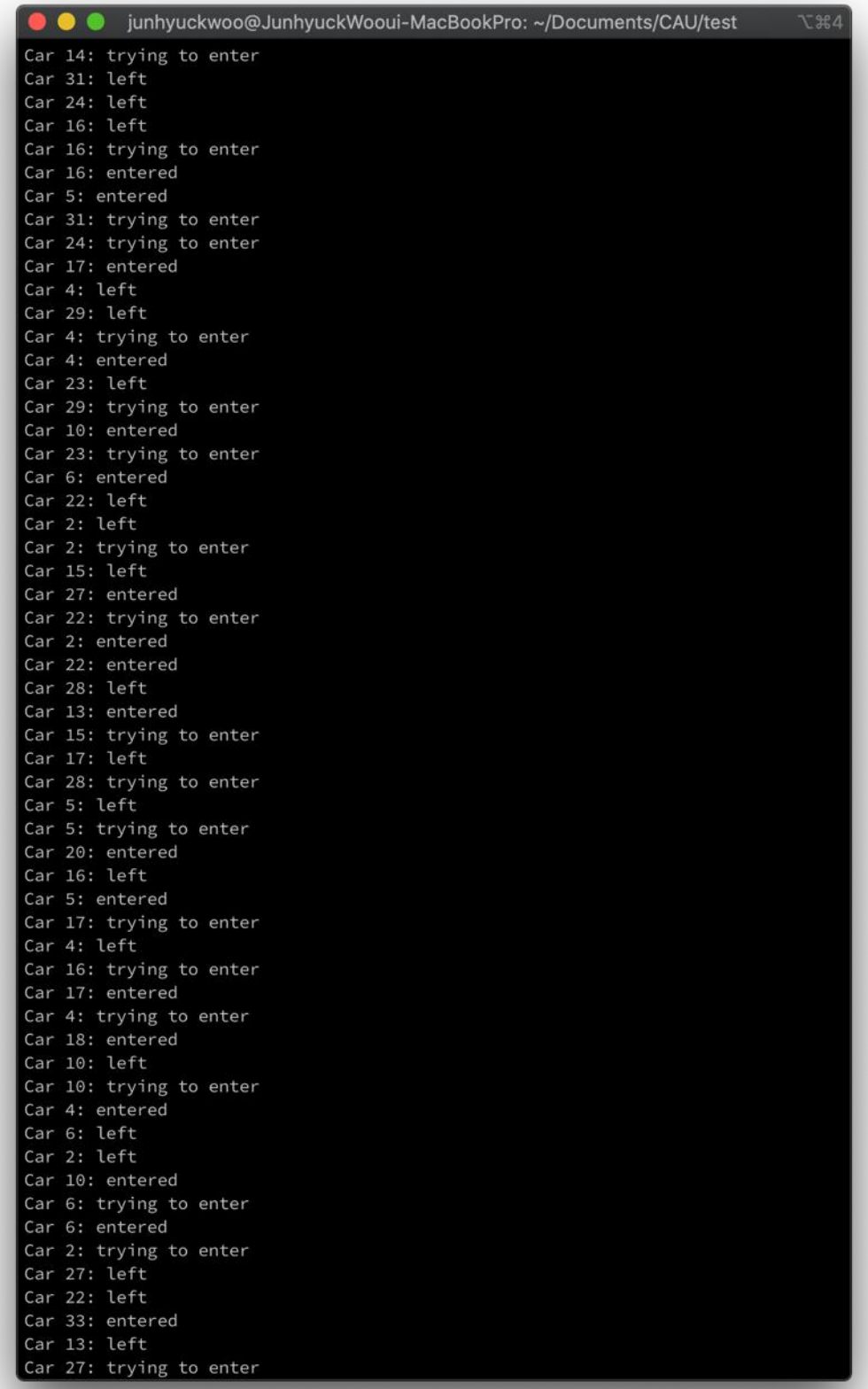
```
// Writer: Junhyuck Woo
// Lecture: Multicore Computing
// Organization: Chung-Ang University
// Deadline: May 17, 2020
// Project #2
// - BlockingQueue Only

import java.util.concurrent.ArrayBlockingQueue;
import java.util.concurrent.BlockingQueue;

public class ParkingGarageSimulation {
    // Main
    public static void main(String[] args) {
        // Create hte blocking queue
        BlockingQueue places = new ArrayBlockingQueue<String>(10);

        // Create the garage
        Garage garage = new Garage(places);

        // Create the Car
        for (int i=1; i<=40; i++){
            Car c = new Car("Car " +i, places, garage);
        }
    }
}

class Garage {

    // Variable
    private BlockingQueue places;

    // Constructor
    Garage(BlockingQueue places) {
        this.places = places;
    }

    public synchronized void enter(String name) {

        try{
            // Car enters the garage
            this.places.put(name);
            synchronized (System.out) {
                // Print the status of car: ENTER
                System.out.println(name + ": entered");
            }
        }
        catch (InterruptedException e) {}
    }

    public void leave() {
        synchronized (System.out) {
            try {
                // Print the status of car: LEAVE
                System.out.println(this.places.take() + ": left");
            }
            catch (InterruptedException e) {}
        }
    }
}


class Car extends Thread {
    // Variables
    private BlockingQueue places;
    private Garage garage;

    // Constructor
    public Car(String name, BlockingQueue places, Garage garage) {
        super(name);
        this.places = places;
        this.garage = garage;
        start();
    }

    // Runner
    public void run() {
        while (true) {
            try {
                sleep((int)Math.random()*10000);
            }
            catch (InterruptedException e) {}
            // Print the status of car: TRY TO ENTER
            System.out.println(getName() + ": trying to enter");
            garage.enter(getName());

            try {
                sleep((int)Math.random()*10000);
            }
            catch (InterruptedException e) {}
            garage.leave();
        }
    }
}
```

OUTPUT - ParkingGarageSimulationPC

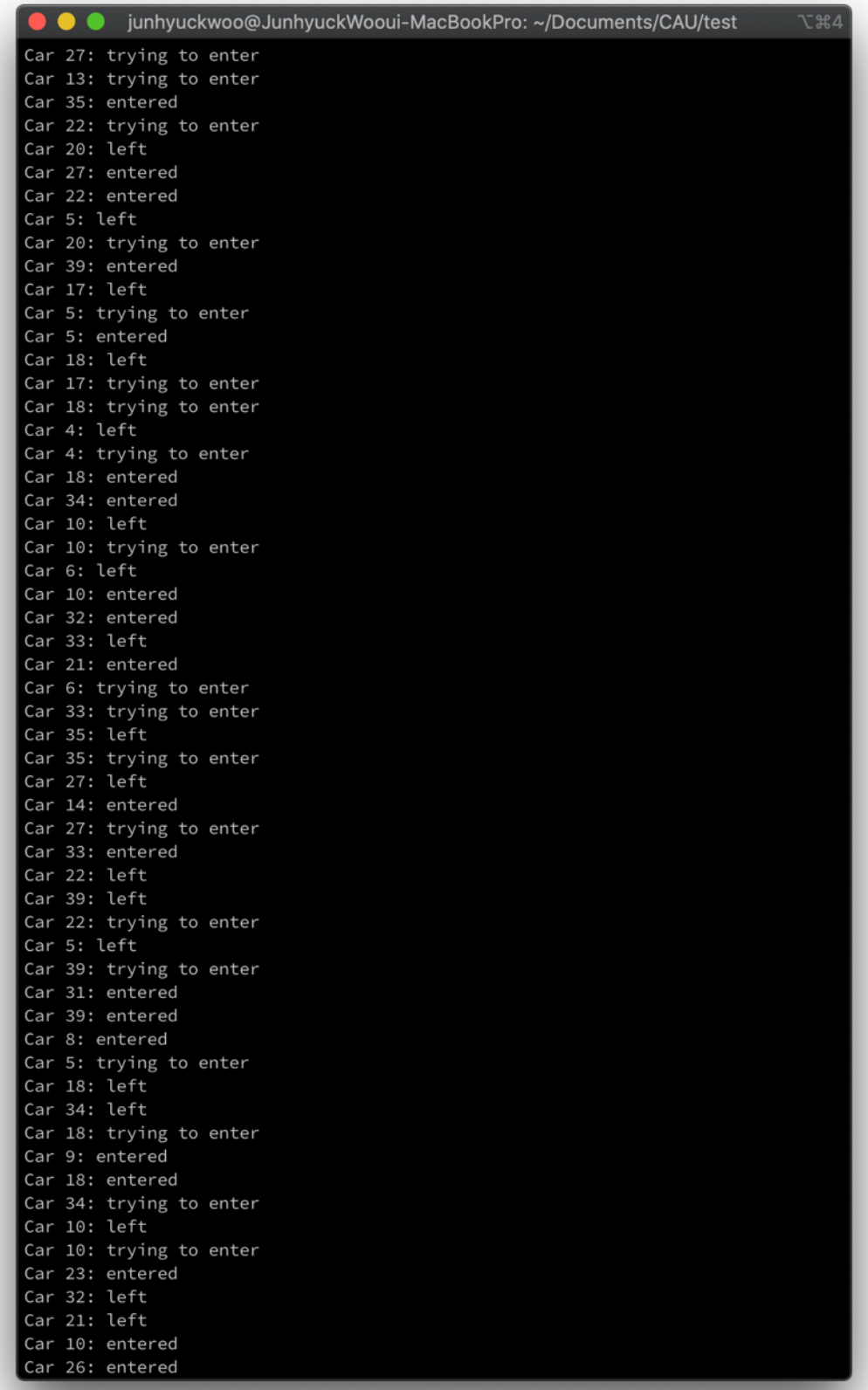
1. Output 1



2. Output 2

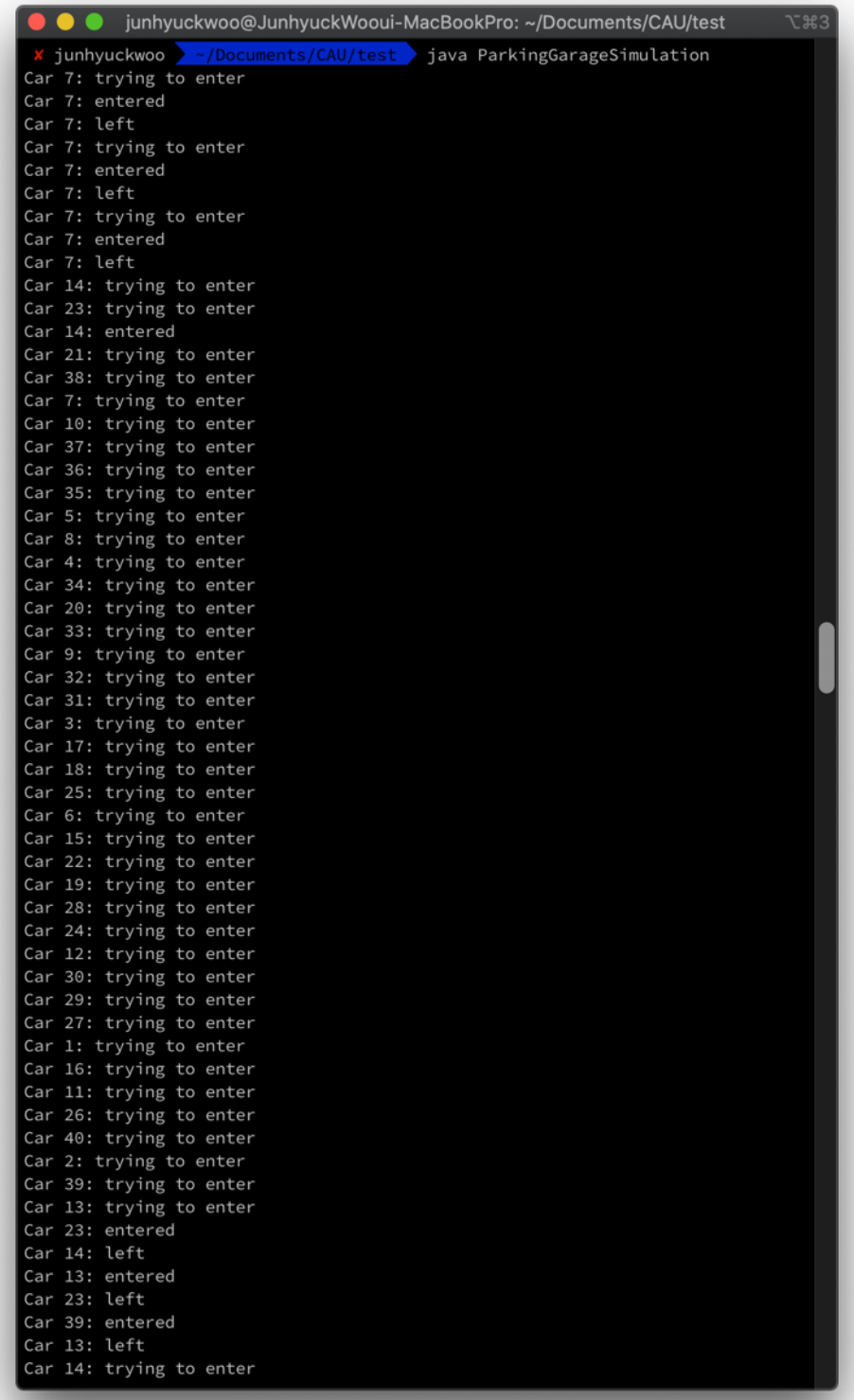

junhyuckwoo@JunhyuckWooui-MacBookPro: ~/Documents/CAU/test
⌥⌘4
Car 14: trying to enter
Car 31: left
Car 24: left
Car 16: left
Car 16: trying to enter
Car 16: entered
Car 5: entered
Car 31: trying to enter
Car 24: trying to enter
Car 17: entered
Car 4: left
Car 29: left
Car 4: trying to enter
Car 4: entered
Car 23: left
Car 29: trying to enter
Car 10: entered
Car 23: trying to enter
Car 6: entered
Car 22: left
Car 2: left
Car 2: trying to enter
Car 15: left
Car 27: entered
Car 22: trying to enter
Car 2: entered
Car 22: entered
Car 28: left
Car 13: entered
Car 15: trying to enter
Car 17: left
Car 28: trying to enter
Car 5: left
Car 5: trying to enter
Car 20: entered
Car 16: left
Car 5: entered
Car 17: trying to enter
Car 4: left
Car 16: trying to enter
Car 17: entered
Car 4: trying to enter
Car 18: entered
Car 10: left
Car 10: trying to enter
Car 4: entered
Car 6: left
Car 2: left
Car 10: entered
Car 6: trying to enter
Car 6: entered
Car 2: trying to enter
Car 27: left
Car 22: left
Car 33: entered
Car 13: left
Car 27: trying to enter

3. Output 3

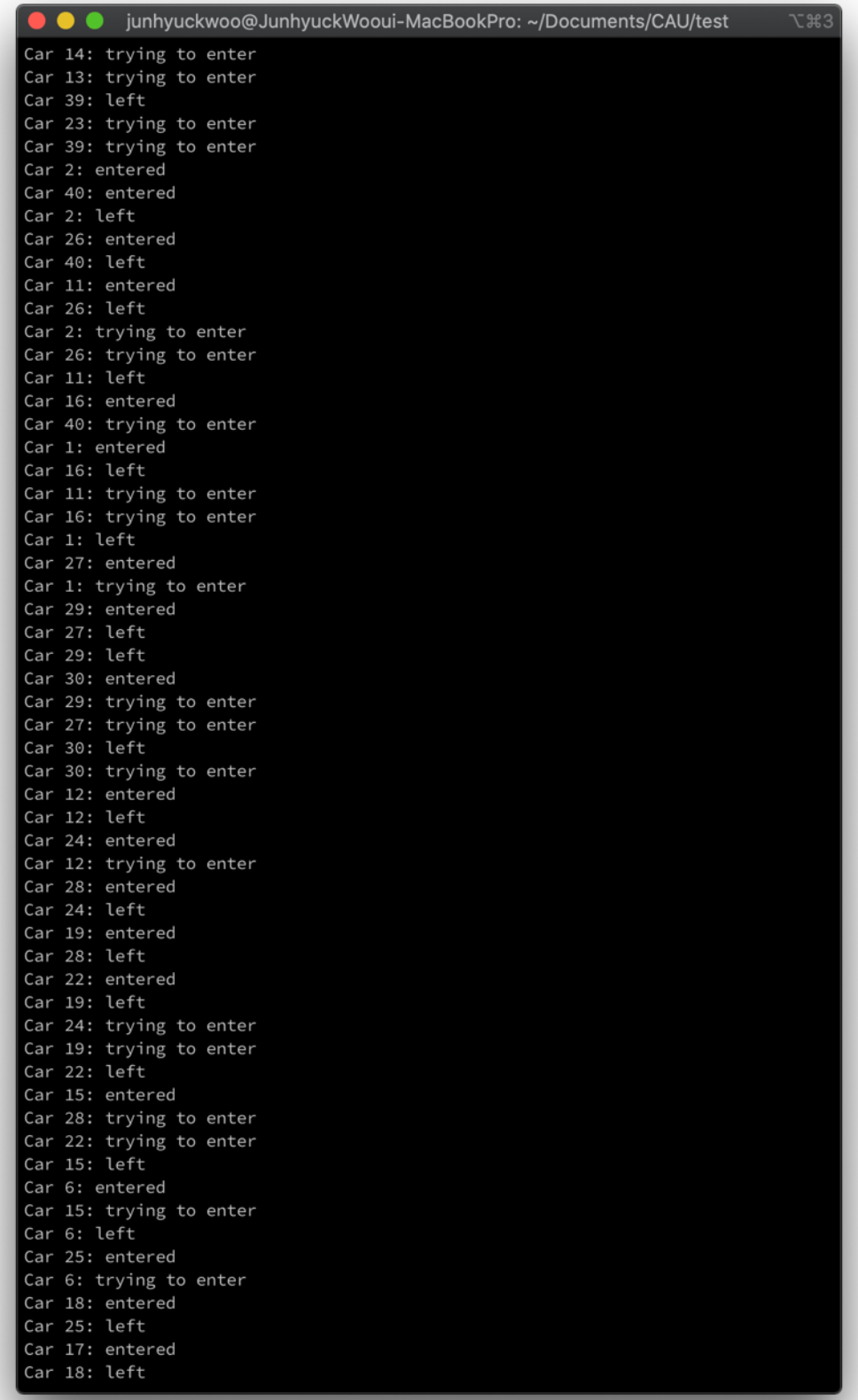

junhyuckwoo@JunhyuckWooui-MacBookPro: ~/Documents/CAU/test ⌥⌘4
Car 27: trying to enter
Car 13: trying to enter
Car 35: entered
Car 22: trying to enter
Car 20: left
Car 27: entered
Car 22: entered
Car 5: left
Car 20: trying to enter
Car 39: entered
Car 17: left
Car 5: trying to enter
Car 5: entered
Car 18: left
Car 17: trying to enter
Car 18: trying to enter
Car 4: left
Car 4: trying to enter
Car 18: entered
Car 34: entered
Car 10: left
Car 10: trying to enter
Car 6: left
Car 10: entered
Car 32: entered
Car 33: left
Car 21: entered
Car 6: trying to enter
Car 33: trying to enter
Car 35: left
Car 35: trying to enter
Car 27: left
Car 14: entered
Car 27: trying to enter
Car 33: entered
Car 22: left
Car 39: left
Car 22: trying to enter
Car 5: left
Car 39: trying to enter
Car 31: entered
Car 39: entered
Car 8: entered
Car 5: trying to enter
Car 18: left
Car 34: left
Car 18: trying to enter
Car 9: entered
Car 18: entered
Car 34: trying to enter
Car 10: left
Car 10: trying to enter
Car 23: entered
Car 32: left
Car 21: left
Car 10: entered
Car 26: entered

OUTPUT - ParkingGarageSimulation

1. Output 1



2. Output 2


junhyuckwoo@JunhyuckWooui-MacBookPro: ~/Documents/CAU/test ⌥⌘3
Car 14: trying to enter
Car 13: trying to enter
Car 39: left
Car 23: trying to enter
Car 39: trying to enter
Car 2: entered
Car 40: entered
Car 2: left
Car 26: entered
Car 40: left
Car 11: entered
Car 26: left
Car 2: trying to enter
Car 26: trying to enter
Car 11: left
Car 16: entered
Car 40: trying to enter
Car 1: entered
Car 16: left
Car 11: trying to enter
Car 16: trying to enter
Car 1: left
Car 27: entered
Car 1: trying to enter
Car 29: entered
Car 27: left
Car 29: left
Car 30: entered
Car 29: trying to enter
Car 27: trying to enter
Car 30: left
Car 30: trying to enter
Car 12: entered
Car 12: left
Car 24: entered
Car 12: trying to enter
Car 28: entered
Car 24: left
Car 19: entered
Car 28: left
Car 22: entered
Car 19: left
Car 24: trying to enter
Car 19: trying to enter
Car 22: left
Car 15: entered
Car 28: trying to enter
Car 22: trying to enter
Car 15: left
Car 6: entered
Car 15: trying to enter
Car 6: left
Car 25: entered
Car 6: trying to enter
Car 18: entered
Car 25: left
Car 17: entered
Car 18: left

3. Output 3


junhyuckwoo@JunhyuckWooui-MacBookPro: ~/Documents/CAU/test ⌥⌘3
Car 18: left
Car 3: entered
Car 17: left
Car 25: trying to enter
Car 17: trying to enter
Car 31: entered
Car 3: left
Car 18: trying to enter
Car 3: trying to enter
Car 32: entered
Car 31: left
Car 32: left
Car 9: entered
Car 32: trying to enter
Car 31: trying to enter
Car 33: entered
Car 9: left
Car 20: entered
Car 33: left
Car 34: entered
Car 20: left
Car 9: trying to enter
Car 20: trying to enter
Car 4: entered
Car 34: left
Car 33: trying to enter
Car 34: trying to enter
Car 8: entered
Car 4: left
Car 5: entered
Car 8: left
Car 35: entered
Car 5: left
Car 4: trying to enter
Car 5: trying to enter
Car 36: entered
Car 35: left
Car 8: trying to enter
Car 35: trying to enter
Car 36: left
Car 37: entered
Car 36: trying to enter
Car 10: entered
Car 37: left
Car 7: entered
Car 37: trying to enter
Car 38: entered
Car 10: left
Car 21: entered
Car 7: left
Car 38: left
Car 21: left
Car 21: trying to enter
Car 38: trying to enter
Car 10: trying to enter
Car 37: entered
Car 36: entered
Car 7: trying to enter