1 /\*\*  
 2 \* This class ParkingLot keeps tracks of cars parked and in queue for a  
 3 \* parking lot. It accepts the following:  
 4 \* 5 digit license plate number -   
 5 \* determines whether or not the car the car is already in the  
 6 \* parking lot or queue  
 7 \* determines if the car has a valid decal to park  
 8 \* determines if the lot is full (sends car to queue) or if it  
 9 \* can park  
 10 \* negative 5 digit license plate number -   
 11 \* determines whether or not the car is in the parking lot or  
 12 \* queue  
 13 \* removes car from queue  
 14 \* puts cars from lot onto street, removes car from lot, puts  
 15 \* cars from street back into lot  
 16 \* 0 - exits the program  
 17 \*  
 18 \* printCars(Stack<Integer>, Queue<Integer<>) method prints out each  
 19 \* car in the lot and queue  
 20 \*  
 21 \* parkCarCheck(Stack<Integer> lot, Queue<Integer> queue, int plate,   
 22 \* int decal, boolean negative) method checks what to do with  
 23 \* the user input information -   
 24 \* return 1 if car is already in lot or queue (if adding), if  
 25 \* car is not in lot or queue (if removing), or if decal is  
 26 \* invalid  
 27 \* return 2 if adding car to lot  
 28 \* return 3 if adding car to queue because lot is full  
 29 \* return 4 if removing car from lot  
 30 \* return 5 if removing car from queue  
 31 \*   
 32 \* @author Stephanie Gremillion  
 33 \* @version 17.0.2  
 34 \* @since 2022/07/25  
 35 \*/  
 36   
 37 import java.util.Random;  
 38 import java.util.Stack;  
 39 import java.util.Queue;  
 40 import java.util.LinkedList;  
 41 import java.util.Scanner;  
 42   
 43 public class ParkingLot {  
 44 public static void main(String[] args) {  
 45 // variables  
 46 Random rand = new Random();  
 47 int plate;  
 48 boolean negative = false;  
 49 int result;  
 50 String temp;  
 51 Scanner input = new Scanner(System.in);  
 52 Stack<Integer> lot = new Stack<>();  
 53 Stack<Integer> street = new Stack<>();  
 54 Queue<Integer> queue = new LinkedList<>();  
 55   
 56 // adding first 10 cars  
 57 for(int i = 0; i < 15; i++) {  
 58 plate = rand.nextInt(90000); // creating a new license plate  
 59 while(lot.contains(plate)) { // if parking lot already has plate, make new plate  
 60 plate = rand.nextInt(90000);  
 61 }  
 62 lot.push(plate); // adding plate to parking lot  
 63 }  
 64   
 65 printCars(lot, queue);  
 66   
 67 do {  
 68 // display  
 69 System.out.println("Please enter your license plate number.");  
 70 System.out.println("Start with - if you wish to remove your car or enter 0 to exit.");  
 71 temp = input.next();  
 72   
 73 // checking if negative  
 74 negative = false;  
 75 if(temp.startsWith("-")) {  
 76 temp = temp.substring(1);  
 77 negative = true;  
 78 }  
 79   
 80 // checking if valid  
 81 if(Integer.parseInt(temp) != 0) {  
 82 while((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) < -99999) || (Integer.parseInt(temp) > 99999) || (temp.length() < 5)) { // error if invalid selection  
 83 System.out.println("Please enter a valid number.");  
 84 temp = input.next();  
 85 negative = false;  
 86 if(temp.startsWith("-")) {  
 87 temp = temp.substring(1);  
 88 negative = true;  
 89 }  
 90 }  
 91 }  
 92 plate = Integer.parseInt(temp);  
 93 result = parkCarCheck(lot, queue, plate, rand.nextInt(), negative); // checking what happens to car  
 94   
 95 switch(result) {  
 96 case 2: // car goes in lot  
 97 lot.push(plate);  
 98 break;  
 99 case 3: // car goes in queue  
100 queue.add(plate);  
101 break;  
102 case 4: // remove car from lot  
103 while(lot.peek() != (plate)) { // putting cars on street  
104 street.push(lot.pop());  
105 }  
106 lot.pop(); // removes car  
107 while(street.size() > 0) {  
108 lot.push(street.pop()); // putting cars back in lot  
109 }  
110   
111 if((lot.size() < 15) && (queue.size() > 0)) { // if room in lot, add from queue  
112 System.out.print("Car " + queue.peek() + " from queue has been added to parking lot.\n");  
113 lot.add(queue.remove());  
114 }  
115 break;  
116 case 5: // remove car from queue  
117 for(int i = 0; i < queue.size(); i++) {  
118 if(queue.peek() == plate) { queue.remove(); }  
119 else { queue.add(queue.poll()); }  
120 }  
121 break;  
122 default:  
123 break;  
124 }  
125 System.out.println();  
126 printCars(lot, queue);  
127 } while(plate != 0);  
128   
129 }  
130   
131 public static void printCars(Stack<Integer> lot, Queue<Integer> queue) {  
132 // printing parking lot  
133 System.out.println("Parking Spaces: " + (15 - lot.size()) + "/15 Spaces Available");  
134 System.out.println("-------------------------------------");  
135 lot.forEach(car -> {  
136 System.out.printf("%05d\n", car);  
137 });  
138 System.out.println();  
139   
140 // printing car queue  
141 System.out.println("Cars in Queue");  
142 System.out.println("-------------");  
143 if(queue.size() == 0) { System.out.println("Empty"); }  
144 else {  
145 for(Integer next : queue) {  
146 System.out.printf("%05d\n", next);  
147 }  
148 }  
149 System.out.println();  
150 }  
151   
152 public static int parkCarCheck(Stack<Integer> lot, Queue<Integer> queue, int plate, int decal, boolean negative) {  
153 int result = 0;  
154   
155 if(plate == 0) { result = 1; } // if exit  
156 else if(negative == false) { // if add car  
157 if(lot.contains(plate)) { // if already in lot  
158 System.out.println("Your car is already parked in the parking lot.\n");  
159 result = 1;  
160 }  
161 else if(queue.contains(plate)) { // if already in queue  
162 System.out.println("Your car is already in the queue.\n");  
163 result = 1;  
164 }  
165 else if(decal % 2 != 0) { // if improper decal  
166 System.out.println("Checking decal...");  
167 System.out.println("Sorry, your car does not have the proper decal for this parking lot.\n");  
168 result = 1;  
169 }  
170 else if(lot.size() < 15) { // if lot has room  
171 System.out.println("Checking decal...");  
172 System.out.println("Your decal is valid. Your car has been parked in the lot.\n");  
173 result = 2;   
174 }  
175 else { // if lot is full  
176 System.out.println("Checking decal...");  
177 System.out.println("Your decal is valid. The parking lot is full. Your car has been added to the queue.\n");  
178 result = 3;   
179 }  
180 }  
181 else { // if remove car  
182 if(lot.contains(plate)) { // remove car from lot  
183 System.out.println("Your car has been removed from the parking lot.\n");  
184 result = 4;   
185 }   
186 else if(queue.contains(plate)) { // remove car from queue  
187 System.out.println("Your car has been removed from the queue.\n");  
188 result = 5;   
189 }   
190 else { // if car is not in lot or queue  
191 System.out.println("Your car is not in the parking lot or queue.\n");  
192 result = 1;  
193 }  
194 }  
195   
196 return result;  
197 }  
198 }