

COMP2401 - Assignment #3

(Due: Mon. Feb 25, 2019 @ 12 noon)

In this assignment, you will gain experience in using **structs**, **pointers**, **malloc()** and **free()**.

Write a program called **parkingSimulator.c** that contains the following **main()** function:

```
int main() {
    Car car1, car2, car3, car4, car5, car6, car7, car8, car9;
    ParkingLot p1, p2;

    // Set up 9 cars
    initializeCar(&car1, "ABC 123", 0);
    initializeCar(&car2, "ABC 124", 0);
    initializeCar(&car3, "ABD 314", 0);
    initializeCar(&car4, "ADE 901", 0);
    initializeCar(&car5, "AFR 304", 0);
    initializeCar(&car6, "AGD 888", 0);
    initializeCar(&car7, "AAA 111", 0);
    initializeCar(&car8, "ABB 001", 0);
    initializeCar(&car9, "XYZ 678", 1);

    // Set up two parking lots
    initializeLot(&p1, 1, 4, 5.5, 20.0);
    initializeLot(&p2, 2, 6, 3.0, 12.0);

    printLotInfo(p1);
    printLotInfo(p2);
    printf("\n");

    // Simulate cars entering the lots
    carEnters(&p1, &car1, 7, 15);
    carEnters(&p1, &car2, 7, 25);
    carEnters(&p2, &car3, 8, 0);
    carEnters(&p2, &car4, 8, 10);
    carEnters(&p1, &car5, 8, 15);
    carEnters(&p1, &car6, 8, 20);
    carEnters(&p1, &car7, 8, 30);
    carEnters(&p2, &car7, 8, 32);
    carEnters(&p2, &car8, 8, 50);
    carEnters(&p2, &car9, 8, 55);

    printf("\n");
    printLotInfo(p1);
    printLotInfo(p2);
    printf("\n");

    // Simulate cars leaving the lots
    carLeaves(&p2, &car4, 9, 0);
    carLeaves(&p1, &car2, 9, 5);
    carLeaves(&p1, &car6, 10, 0);
    carLeaves(&p1, &car1, 10, 30);
    carLeaves(&p2, &car8, 13, 0);
    carLeaves(&p2, &car9, 15, 15);
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

```

carEnters(&p1, &car8, 17, 10);
carLeaves(&p1, &car5, 17, 50);
carLeaves(&p2, &car7, 18, 0);
carLeaves(&p2, &car3, 18, 15);
carLeaves(&p1, &car8, 20, 55);

printf("\n");
printLotInfo(p1);
printLotInfo(p2);
printf("\n");

// Display the total revenue
printf("Total revenue of Lot 1 is $%4.2f\n", p1.revenue);
printf("Total revenue of Lot 2 is $%4.2f\n", p2.revenue);
}

```

This code simulates a few cars parking in two parking lots. To make it work, you will have to follow the directions below carefully. You MAY NOT alter the **main()** function code in any way. Follow the steps below and pay attention to the clear direction as to parameter types and what each function should do. Make sure to use comments in your code and make sure that it is neat and organized. Just to make one thing clear ... even though we will have multiple cars parking in the parking lots ... at this point the parking lot does NOT keep an array of cars.

- Create a **struct** called **Time** that stores an **hour** and **minute** amount. This could be used to represent a time of day or an amount of time that has elapsed.



- Create a **struct** called **Car** that stores a **plateNumber** for a car (which must be a char pointer). A single character which indicates whether or not the car has a **permit**, an **enteringTime** which is a **Time** struct that indicates the time that the car entered the lot, and a **lotParkedIn** number which is an **int** indicating the number of the lot that the car is parked in.



- Create a **struct** called **ParkingLot** that stores a **lotNumber** (as an integer) to represent it's unique ID, an **hourlyRate** (a double) representing the amount that a car owner would pay per hour to be parked in that lot, a **maxCharge** (a double) indicating the maximum amount that the car owner would pay to park the car for the day, an integer **capacity** representing the maximum number of cars that can park in that lot, a **currentCarCount** indicating the number of cars currently parked in the lot, and finally a **revenue** (a double) that keeps track of the amount of money taken in by that lot so far.



- Create the following **Time struct** related functions:

```

// Sets the hours and minutes amount for the given time t based
// on the specified hours h. (e.g., 1.25 hours would be 1 hour
// and 15 minutes)
void setHours(Time *t, double h) { ... }

// Takes two Time objects (not pointers) and computes the difference
// in time from t1 to t2 and then stores that difference in the diff
// Time (which must be a pointer)
void difference(Time t1, Time t2, Time *diff) { ... }

```

- Create the following **Car struct** related functions:

```
// Initialize the car pointed to by c to have the given plate and
// hasPermit status. The car should have its lotParkedIn set to
// 0 and enteringTime to be -1 hours and -1 minutes.
void initializeCar(Car *c, char *plate, char hasPermit) { ... }
```

- Create the following **ParkingLot struct** related functions:

```
// Initialize the lot pointed to by p to have the given number,
// capacity, hourly rate and max charge values. The currentCarCount
// and revenue should be at 0.
void initializeLot(ParkingLot *p, int num, int cap,
                 double rate, double max) { ... }
```

```
// Print out the parking lot parameters so that it displays as
// follows:
// Parking Lot #2 - rate = $3.00, capacity 6, current cars 5
void printLotInfo(ParkingLot p) { ... }
```

- Add appropriate functions called **carEnters()** and **carLeaves()** so that the main method above works correctly and produces the following output results exactly:

```
Parking Lot #1 - rate = $5.50, capacity 4, current cars 0
Parking Lot #2 - rate = $3.00, capacity 6, current cars 0
```

```
Car ABC 123 enters Lot 1 at 7:15.
Car ABC 124 enters Lot 1 at 7:25.
Car ABD 314 enters Lot 2 at 8:00.
Car ADE 901 enters Lot 2 at 8:10.
Car AFR 304 enters Lot 1 at 8:15.
Car AGD 888 enters Lot 1 at 8:30.
Car AAA 111 arrives at Lot 1 at 8:30, but the lot is full.
Car AAA 111 cannot get in.
Car AAA 111 enters Lot 2 at 8:32.
Car ABB 001 enters Lot 2 at 8:50.
Car XYZ 678 enters Lot 2 at 8:55.
```

```
Parking Lot #1 - rate = $5.50, capacity 4, current cars 4
Parking Lot #2 - rate = $3.00, capacity 6, current cars 5
```

```
Car ADE 901 leaves Lot 2 at 9:00 paid $3.00.
Car ABC 124 leaves Lot 1 at 9:05 paid $11.00.
Car AGD 888 leaves Lot 1 at 10:00 paid $11.00.
Car ABC 123 leaves Lot 1 at 10:30 paid $20.00.
Car ABB 001 leaves Lot 2 at 13:00 paid $12.00.
Car XYZ 678 leaves Lot 2 at 15:15.
Car ABB 001 enters Lot 1 at 17:10.
Car AFR 304 leaves Lot 1 at 17:50 paid $20.00.
Car AAA 111 leaves Lot 2 at 18:00 paid $12.00.
Car ABD 314 leaves Lot 2 at 18:15 paid $12.00.
Car ABB 001 leaves Lot 1 at 20:55 paid $20.00.
```

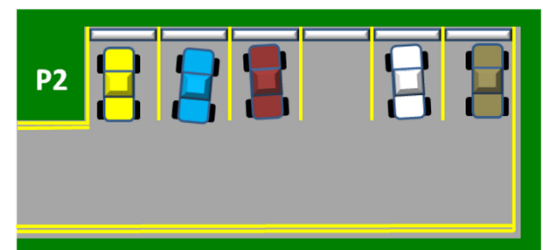
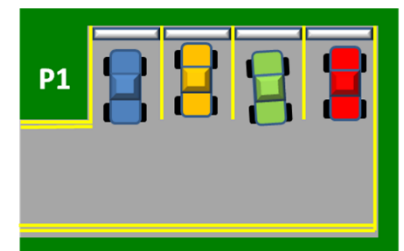
```
Parking Lot #1 - rate = $5.50, capacity 4, current cars 0
Parking Lot #2 - rate = $3.00, capacity 6, current cars 0
```

```
Total revenue of Lot 1 is $82.00
Total revenue of Lot 2 is $39.00
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



We will now modify the code so as to allow an arbitrary number of cars and parking lots to be used by using dynamic memory allocation. Copy your **parkingSimulator.c** program to a file called **simulator2.c**. You will now modify the **simulator2.c** program as explained here. You must complete this part of the assignment WITHOUT altering any of the structs that you defined earlier and you MUST NOT alter ANY of the functions that you wrote. You will just create two new functions and then alter the **main()** function.

- Create a function called **randomPlate()** that takes no parameters and returns a **char ***. The function should generate a random license plate with the format "**xxx ###**" where **X** is a random character from A to Z and **#** is a random digit from 0 to 9. The random string must be dynamically-allocated and a pointer to this string should be returned.
- Create a function called **randomCar()** that takes no parameters and returns a **Car ***. You MUST make use of the **randomPlate()** and **initializeCar()** functions. The pointer to the new dynamically-allocated **Car** should be returned.
- Erase the contents of the **main()** function. Instead, write the main function code so that it follows the directions (in order) as shown below:
 1. Make an array to hold pointers to **50 Cars** and an array to hold pointers to **5 ParkingLots**
 2. Using your **randomPlate()** and **randomCar()** functions, set up **50 cars** with random plates and random permit status and display them.
 3. Set up **5** dynamically-allocated parking lots with numbers **1, 2, 3, 4, 5**; capacities **5, 10, 15, 20, 25**; hourlyRates **\$4, \$5, \$6, \$7, \$8**; and max charges **\$12, \$14, \$16, \$18, \$20**.
 4. Display the lot info for all parking lots
 5. Simulate all **50 cars** trying to enter a randomly-chosen lot starting at **6am** such that each car enters exactly 5 minutes after the previous car entered. Make sure that each car attempts to enter a lot by calling the **carEnters()** function.
 6. Display the lot info again for all parking lots.
 7. Simulate all parked cars trying to leave the lots (use **carLeaves()**) starting at **11am** such that each car leaves exactly 5 minutes after the previous car left. Make sure that a car has already parked before you try to simulate it leaving.
 8. Display the lot info again for all parking lots.
 9. Display the revenue for all parking lots.
 10. Free up ALL memory that you had dynamically allocated. Use **valgrind** to ensure that everything was freed properly
- Ensure that each time you run your code, you get different random values.

Here is an example of the output that you should get:

```
Car JAW 283 with permit 0
Car WSZ 290 with permit 0
Car QND 508 with permit 1
Car COL 466 with permit 1
Car VAD 100 with permit 1
Car XKL 703 with permit 0
Car BLU 699 with permit 0
Car BTC 135 with permit 0
```

Car YEP 717 with permit 1
Car FSJ 117 with permit 1
Car FEY 267 with permit 1
Car PRF 632 with permit 1
Car SUK 690 with permit 1
Car ESJ 347 with permit 0
Car ONK 763 with permit 1
Car ICZ 971 with permit 0
Car DLJ 828 with permit 1
Car GVX 452 with permit 0
Car ZYI 547 with permit 1
Car DCH 412 with permit 0
Car XKZ 083 with permit 0
Car DEK 392 with permit 1
Car NNP 449 with permit 1
Car YRH 075 with permit 0
Car YVY 829 with permit 1
Car DHT 241 with permit 1
Car MLL 990 with permit 1
Car MZM 467 with permit 0
Car KJB 319 with permit 0
Car LYA 527 with permit 0
Car SYM 139 with permit 1
Car JZQ 891 with permit 0
Car OWT 927 with permit 0
Car KUK 874 with permit 0
Car YEG 617 with permit 0
Car NRA 866 with permit 1
Car OTZ 167 with permit 1
Car WNJ 227 with permit 0
Car AEO 938 with permit 0
Car MON 025 with permit 1
Car WFR 496 with permit 1
Car PJR 480 with permit 1
Car EUW 194 with permit 1
Car IGU 883 with permit 0
Car BWT 903 with permit 1
Car BAZ 646 with permit 0
Car IRX 447 with permit 0
Car Kfv 746 with permit 1
Car HYI 380 with permit 1
Car XLR 946 with permit 0

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Parking Lot #1 - rate = \$4.00, capacity 5, current cars 0
Parking Lot #2 - rate = \$5.00, capacity 10, current cars 0
Parking Lot #3 - rate = \$6.00, capacity 15, current cars 0
Parking Lot #4 - rate = \$7.00, capacity 20, current cars 0
Parking Lot #5 - rate = \$8.00, capacity 25, current cars 0

Car JAW 283 enters Lot 4 at 6:00.
Car WSZ 290 enters Lot 5 at 6:05.
Car QND 508 enters Lot 2 at 6:10.
Car COL 466 enters Lot 4 at 6:15.
Car VAD 100 enters Lot 2 at 6:20.
Car XKL 703 enters Lot 5 at 6:25.
Car BLU 699 enters Lot 4 at 6:30.
Car BTC 135 enters Lot 4 at 6:35.
Car YEP 717 enters Lot 2 at 6:40.
Car FSJ 117 enters Lot 3 at 6:45.
Car FEY 267 enters Lot 2 at 6:50.
Car PRF 632 enters Lot 4 at 6:55.
Car SUK 690 enters Lot 4 at 7:00.
Car ESJ 347 enters Lot 2 at 7:05.
Car ONK 763 enters Lot 3 at 7:10.
Car ICZ 971 enters Lot 1 at 7:15.
Car DLJ 828 enters Lot 5 at 7:20.
Car GVX 452 enters Lot 1 at 7:25.
Car ZYI 547 enters Lot 3 at 7:30.
Car DCH 412 enters Lot 5 at 7:35.
Car XKZ 083 enters Lot 2 at 7:40.

Car DEK 392 enters Lot 4 at 7:45.
Car NNP 449 enters Lot 4 at 7:50.
Car YRH 075 enters Lot 3 at 7:55.
Car YVY 829 enters Lot 1 at 8:00.
Car DHT 241 enters Lot 3 at 8:05.
Car MLL 990 enters Lot 5 at 8:10.
Car MZM 467 enters Lot 4 at 8:15.
Car KJB 319 enters Lot 3 at 8:20.
Car LYA 527 enters Lot 2 at 8:25.
Car SYM 139 enters Lot 2 at 8:30.
Car JZQ 891 enters Lot 1 at 8:35.
Car OWT 927 enters Lot 2 at 8:40.
Car KUK 874 enters Lot 3 at 8:45.
Car YEG 617 enters Lot 5 at 8:50.
Car NRA 866 enters Lot 3 at 8:55.
Car OTZ 167 enters Lot 3 at 9:00.
Car WNJ 227 enters Lot 3 at 9:05.
Car AEO 938 enters Lot 2 at 9:10.
Car MON 025 enters Lot 5 at 9:15.
Car WFR 496 enters Lot 5 at 9:20.
Car PJR 480 enters Lot 4 at 9:25.
Car EUW 194 enters Lot 4 at 9:30.
Car IGU 883 enters Lot 3 at 9:35.
Car BWT 903 enters Lot 5 at 9:40.
Car BAZ 646 enters Lot 1 at 9:45.
Car IRX 447 enters Lot 4 at 9:50.
Car Kfv 746 enters Lot 4 at 9:55.
Car HYI 380 arrives at Lot 1 at 10:00, but the lot is full.
Car HYI 380 cannot get in.
Car XLR 946 arrives at Lot 1 at 10:05, but the lot is full.
Car XLR 946 cannot get in.

Parking Lot #1 - rate = \$4.00, capacity 5, current cars 5
Parking Lot #2 - rate = \$5.00, capacity 10, current cars 10
Parking Lot #3 - rate = \$6.00, capacity 15, current cars 15
Parking Lot #4 - rate = \$7.00, capacity 20, current cars 13
Parking Lot #5 - rate = \$8.00, capacity 25, current cars 9

Car JAW 283 leaves Lot 4 at 11:00 paid \$16.00.
Car WSZ 290 leaves Lot 5 at 11:05 paid \$18.00.
Car QND 508 leaves Lot 2 at 11:10.
Car COL 466 leaves Lot 4 at 11:15.
Car VAD 100 leaves Lot 2 at 11:20.
Car XKL 703 leaves Lot 5 at 11:25 paid \$18.00.
Car BLU 699 leaves Lot 4 at 11:30 paid \$16.00.
Car BTC 135 leaves Lot 4 at 11:35 paid \$16.00.
Car YEP 717 leaves Lot 2 at 11:40.
Car FSJ 117 leaves Lot 3 at 11:45.
Car FEY 267 leaves Lot 2 at 11:50.
Car PRF 632 leaves Lot 4 at 11:55.
Car SUK 690 leaves Lot 4 at 12:00.
Car ESJ 347 leaves Lot 2 at 12:05 paid \$12.00.
Car ONK 763 leaves Lot 3 at 12:10.
Car ICZ 971 leaves Lot 1 at 12:15 paid \$10.00.
Car DLJ 828 leaves Lot 5 at 12:20.
Car GVX 452 leaves Lot 1 at 12:25 paid \$10.00.
Car ZYI 547 leaves Lot 3 at 12:30.
Car DCH 412 leaves Lot 5 at 12:35 paid \$18.00.
Car XKZ 083 leaves Lot 2 at 12:40 paid \$12.00.
Car DEK 392 leaves Lot 4 at 12:45.
Car NNP 449 leaves Lot 4 at 12:50.
Car YRH 075 leaves Lot 3 at 12:55 paid \$14.00.
Car YVY 829 leaves Lot 1 at 13:00.
Car DHT 241 leaves Lot 3 at 13:05.
Car MLL 990 leaves Lot 5 at 13:10.
Car MZM 467 leaves Lot 4 at 13:15 paid \$16.00.
Car KJB 319 leaves Lot 3 at 13:20 paid \$14.00.
Car LYA 527 leaves Lot 2 at 13:25 paid \$12.00.
Car SYM 139 leaves Lot 2 at 13:30.
Car JZQ 891 leaves Lot 1 at 13:35 paid \$10.00.

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Car OWT 927 leaves Lot 2 at 13:40 paid \$12.00.
Car KUK 874 leaves Lot 3 at 13:45 paid \$14.00.
Car YEG 617 leaves Lot 5 at 13:50 paid \$18.00.
Car NRA 866 leaves Lot 3 at 13:55.
Car OTZ 167 leaves Lot 3 at 14:00.
Car WNJ 227 leaves Lot 3 at 14:05 paid \$14.00.
Car AEO 938 leaves Lot 2 at 14:10 paid \$12.00.
Car MON 025 leaves Lot 5 at 14:15.
Car WFR 496 leaves Lot 5 at 14:20.
Car PJR 480 leaves Lot 4 at 14:25.
Car EUW 194 leaves Lot 4 at 14:30.
Car IGU 883 leaves Lot 3 at 14:35 paid \$14.00.
Car BWT 903 leaves Lot 5 at 14:40.
Car BAZ 646 leaves Lot 1 at 14:45 paid \$10.00.
Car IRX 447 leaves Lot 4 at 14:50 paid \$16.00.
Car Kfv 746 leaves Lot 4 at 14:55.

Parking Lot #1 - rate = \$4.00, capacity 5, current cars 0
Parking Lot #2 - rate = \$5.00, capacity 10, current cars 0
Parking Lot #3 - rate = \$6.00, capacity 15, current cars 0
Parking Lot #4 - rate = \$7.00, capacity 20, current cars 0
Parking Lot #5 - rate = \$8.00, capacity 25, current cars 0

Total revenue of Lot 1 is \$40.00
Total revenue of Lot 2 is \$60.00
Total revenue of Lot 3 is \$70.00
Total revenue of Lot 4 is \$80.00
Total revenue of Lot 5 is \$72.00

Assignment Project Exam Help

IMPORTANT SUBMISSION INSTRUCTIONS:

Submit all of your **c source code** files as a single tar file containing:

1. A **Readme** text file containing
 - your name and studentNumber
 - a list of source files submitted
 - any specific instructions for compiling and/or running your code
2. All of your **.c source** files and all other files needed for testing/running your programs.
3. Any output files required, if there are any.

The code **MUST** compile and run on the course VM, which is **COMP2404B-W19**.

- If your internet connection at home is down or does not work, we will not accept this as a reason for handing in an assignment late ... so make sure to submit the assignment WELL BEFORE it is due !
 - You WILL lose marks on this assignment if any of your files are missing. So, make sure that you hand in the correct files and version of your assignment. You will also lose marks if your code is not **written neatly with proper indentation and containing a reasonable number of comments**. See course notes for examples of what is proper indentation, writing style and reasonable commenting).
-