CAR PARKING SYSTEM

```
/* BG1221 Computer Programming For Engineering, 1/2018

Group Member:
1. Kasidis (Ken) Arunruangsirilert ID 6116803 Section 641
2. Thanyawee (Meimei) Chunhasawasdikul ID 6115300 Section 641

Project Due 26th November 2018
Just Another Car Parking Management System by KenSoftTH */
```

WHY CAR PARKING PROJECT?

Glurr

จอดรถซ้อนคันของคุณ mazda

คุณอาจจะไม่เห็น post นี้แต่ถ้าใครรู้เจ้าของก็ฝากบอกด้วยนะว่าเป็นการจอดที่ไร้มารยาทสุดๆ ไม่ได้จะประจานนะ อาจจะเป็นมือใหม่แต่ควรรู้ว่ารถตัวเองเข้าตัว N ยังไงด้วย มันเสียเวลาและอารมณ์มากๆๆๆๆ สร้างความเดือดร้อนให้พี่ยามอะใช้ตั้ง 6 คนเลยนะ แล้วรถก็ไม่ใช่เล็กๆอะ สงสารพี่ยามที่ต้องมาเข็นรถและเหนื่อยกับความมักง่ายของคุณmazda

MAIN FUNCTION

```
void main() {
    // Declare variables
    int *time;
    int selection = 0;
    int countTransfer = 0;
    time = getTime(); // Call get time function to get time
    configIO(); // Config IO function
    system("cls"); // clear console
    //Copy all the value read from file or input by user to another array
    //The new array will act as a count for the remaining space
    for (countTransfer = 0; countTransfer < 100; countTransfer++) {
        parkingLotLeft[countTransfer] = parkingLotRead[countTransfer];
    }
    //Menu
    while (selection != 5) {
        selection = menu();
        switch (selection) {
        case 1:
            carIn();
            break;
        case 2:
            carOut();
            break;
        case 3:
            displayParkingLots();
            break;
        case 4:
            changeFee();
            break;
        default:
            writeLog();
            break;
        }
    }
}
```

CONFIG IO FUNCTION

```
void configIO() {
    int floor;
    int *parkingLotNumber;

    // This function read and write the config file
    // printf("%d", checkConfigExist());
    if (checkConfigExist()) {
        // If config file exist --> Not the first time
        int count = 0,placeholder;
        printf("File Existed!\n"); // Just for debugging
        FILE *file;
        file = fopen("config.txt", "r"); // Open file for reading
        // Read all 100 elements from the files
        for (count = 0; count < 100; count++) {
            // For the first line, read the number of floor
            if (count == 0) {
                fscanf(file, "Floor=%d\n", &parkingLotRead[0]);
                printf("%d floors\n", parkingLotRead[0]);
            }
            // For the rest, read and store in array
            else {
                fscanf(file, "Floor %d=%d\n", &placeholder, &parkingLotRead[count]);
                printf("Read line %d value %d\n", count + 1, parkingLotRead[count]);
            }
        }
        //Read Parking Fee rate, then store it into an array and print it for debugging
        fscanf(file, "Rate=%d\n", &parkingRate[10][0]);
        printf("%d rates\n", parkingRate[10][0]);
         for (countRate = 0; countRate < 10; countRate++) {
                fscanf(file, "R %d %d\n", &parkingRate[countRate][0], &parkingRate[countRate][1]);
                printf("Read line %d value %d minutes %d baht per hour\n", countRate + 1, parkingRate[countRate][0], parkingRate[countRate][1]);

         }
         // Close the file
        fclose(file);
    }
```

```
t *parkingLotNumber;

This function read and write the config file
printf("%d", checkConfigExist());
(checkConfigExist()) {
 // If config file exist --> Not the first time
 int count = 0,placeholder;
 printf("File Existed!\n"); // Just for debugging
 FILE *file;
 file = fopen("config.txt", "r"); // Open file for reading
 // Read all 100 elements from the files
 for (count = 0; count < 100; count++) {
     // For the first line, read the number of floor
     if (count == 0) {
         fscanf(file, "Floor=%d\n", &parkingLotRead[0]);
         printf("%d floors\n", parkingLotRead[0]);
     }
     // For the rest, read and store in array
     else {
         fscanf(file, "Floor %d=%d\n", &placeholder, &parkingLotRead[count]);
         printf("Read line %d value %d\n", count + 1, parkingLotRead[count]);
     }
 }
 //Read Parking Fee rate, then store it into an array and print it for debugging
 fscanf(file, "Rate=%d\n", &parkingRate[10][0]);
```

ruangsirilert, 6 days ago | 1 author, 23 changes

```
d
ted: Project: CarParkingSystem, Configuration: Debug Win32 ------
vcxproj -> C:\Users\ken15\Documents\Visual Studio 2017\Projects\BG1221ComputerProgrammingForEngineering\BG1221Project\Debug\CarParkingSystem.exe
vcxproj -> C:\Users\ken15\Documents\Visual Studio 2017\Projects\BG1221ComputerProgrammingForEngineering\BG1221Project\Debug\CarParkingSystem.pdb (Partial PDB)
 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

CONFIG IO FUNCTION

```
else {
    // Config file not exist, so launch the setup process!
    // Declare Variable
    int countTransfer = 0;
    // Debugger
    printf("File not Exist!\n");
    // Call two Function to setup
    parkingLotNumber = parkingLotsNumberSetup();
    parkingFeeRateSetup(parkingRate);
    // Pause the console
    system("pause");
    // Write Config to File
    FILE *file;
    file = fopen("config.txt", "w");
    fprintf(file, "Floor=%d\n", parkingLotNumber[0]);
    for (count = 1; count <= 99; count++) {
        fprintf(file,"Floor %d=%d\n",count, parkingLotNumber[count]);
    }
    fprintf(file, "Rate=%d\n", parkingRate[10][0]);
    for (countRate = 0; countRate <= 9; countRate++) {
        fprintf(file, "R %d %d\n", parkingRate[countRate][0], parkingRate[countRate][1]);
    }
    fclose(file);
    for (countTransfer = 0; countTransfer < 100; countTransfer++) {
        parkingLotRead[countTransfer] = parkingLotNumber[countTransfer];
    }
    printf("\nSetup Completed!\n");
}
```

PARKING LOTS NUMBER SETUP FUNCTION

```c
int parkingLotsNumberSetup() {
    // This function receive the number of parking lots and floor from the user
    int floor, count;
    static int parkingLotNumber[100];
    // printf the Menu
    printf("\n---Parking Lots Number Setup Wizard---\n");
    printf("How many floor does the parking building has? :");
    scanf("%d", &floor);
    // Error Checking
    while (floor > 99 || floor < 1) {
        printf("Invalid Input!!! How many floor does the parking building has? :");
        scanf("%d", &floor);
    }
    // Loop scanf
    for (count = 1; count <= floor; count++) {
        printf("Enter number of parking lots for the floor number %d :", count);
        scanf("%d", &parkingLotNumber[count]);
    }
    parkingLotNumber[0] = floor;
    // Return the Value
    return parkingLotNumber;
}
```

System C:\WINDOWS\system32\cmd.exe (Global Scope)

```c
    fclose(file);
}
// Main Function
void main() {
    int *time;
    int selection = 0;
    int countTransfer = 0;
    time = getTime();

    configIO();
    system("cls");
    for (countTransfer = 0; countTransfer < 100;
```

PARKING FEE RATE SETUP FUNCTION

```c
int parkingFeeRateSetup(int rate[11][2]) {
    // This function receive parking fee rate from the user
    // Declare Variables
    int rateCount, count, hour, displayLoop;
    // printf the menu
    printf("\n");
    printf("---Parking Fee Rate Setup Wizard---\n");
    printf("How many parking fee rate are there? (including free parking) :");
    scanf("%d", &rateCount);
    rate[10][0] = rateCount;
    // Check for Invalid Data
    while (rateCount < 1 || rateCount>10) {
        printf("Incorrect Input!! How many parking fee rate are there? (including free parking) :");
        scanf("%d", &rateCount);
    }
```

```
File not Exist!

---Parking Lots Number Setup Wizard---
How many floor does the parking building has? :6
Enter number of parking lots for the floor number 1 :10
Enter number of parking lots for the floor number 2 :10
Enter number of parking lots for the floor number 3 :10
Enter number of parking lots for the floor number 4 :10
Enter number of parking lots for the floor number 5 :10
Enter number of parking lots for the floor number 6 :10

---Parking Fee Rate Setup Wizard---
How many parking fee rate are there? (including free parking) :
```

```c
    // loop scanf
    for (count = 0; count < rateCount; count++) {
        // First time --> Free Parking
        if (count == 0) {
            printf("How much time (in minutes) can people park without any fee?:");
            scanf("%d", &rate[0][0]);
            rate[0][1] = 0;
            printf("%-15s %-15s\n", "Time (Minutes)", "Fee per hour");
            printf("%-15d %-15d\n", rate[0][0], rate[0][1]);
        }
        // Next time is a normal rate
        else {
            printf("\nHow many hours from the start does this rate apply?:");
            scanf("%d", &hour);
            rate[count][0] = hour*60;
            printf("How much fee per hour?:");
            scanf("%d", &rate[count][1]);
            printf("%-15s %-15s\n", "Time (Minutes)", "Fee per hour");
            for (displayLoop=0; displayLoop <= count; displayLoop++) {
                printf("%-15d %-15d\n", rate[displayLoop][0], rate[displayLoop][1]);
            }
        }

    }
    return 0;
```

```
Enter number of parking lots for the floor number 3 :10
Enter number of parking lots for the floor number 4 :10
Enter number of parking lots for the floor number 5 :10
Enter number of parking lots for the floor number 6 :10

---Parking Fee Rate Setup Wizard---
How many parking fee rate are there? (including free parking) :20
Incorrect Input!! How many parking fee rate are there? (including free parking) :20
Incorrect Input!! How many parking fee rate are there? (including free parking) :20
Incorrect Input!! How many parking fee rate are there? (including free parking) :20
Incorrect Input!! How many parking fee rate are there? (including free parking) :3
How much time (in minutes) can people park without any fee?:
```

MAIN FUNCTION

```
void main() {
    // Declare variables
    int *time;
    int selection = 0;
    int countTransfer = 0;
    time = getTime(); // Call get time function to get time
    configIO(); // Config IO function
    system("cls"); // clear console
    //Copy all the value read from file or input by user to another array
    //The new array will act as a count for the remaining space
    for (countTransfer = 0; countTransfer < 100; countTransfer++) {
        parkingLotLeft[countTransfer] = parkingLotRead[countTransfer];
    }
    //Menu
    while (selection != 5) {
        selection = menu();
        switch (selection) {
        case 1:
            carIn();
            break;
        case 2:
            carOut();
            break;
        case 3:
            displayParkingLots();
            break;
        case 4:
            changeFee();
            break;
        default:
            writeLog();
            break;
        }
    }
}
```

MENU FUNCTION

```c
int menu() {
    int selection;
    // printf the menu
    printf("=======================================Just another parking system by KenSoftTH=======================================\n");
    printf("Please select from the menu below\n");
    printf("[1] Car In\n");
    printf("[2] Car Out\n");
    printf("[3] Display/Change Parking Lots\n");
    printf("[4] Display/Change Parking Fee\n");
    printf("[5] End the Day and Print the Report (Exit)\n");
    printf(":: ");
    // scanf the user selection
    scanf("%d", &selection);
    // Check for invalid selection
    while (selection < 1 || selection > 5) {
        printf("Invalid Selection!!\n");
        printf(":: ");
        scanf("%d", &selection);
    }
    return selection;
}
```

```
C:\WINDOWS\system32\cmd.exe
=======================================Just another parking system by KenSoftTH=======================================
Please select from the menu below
[1] Car In
[2] Car Out
[3] Display/Change Parking Lots
[4] Display/Change Parking Fee
[5] End the Day and Print the Report (Exit)
::
```

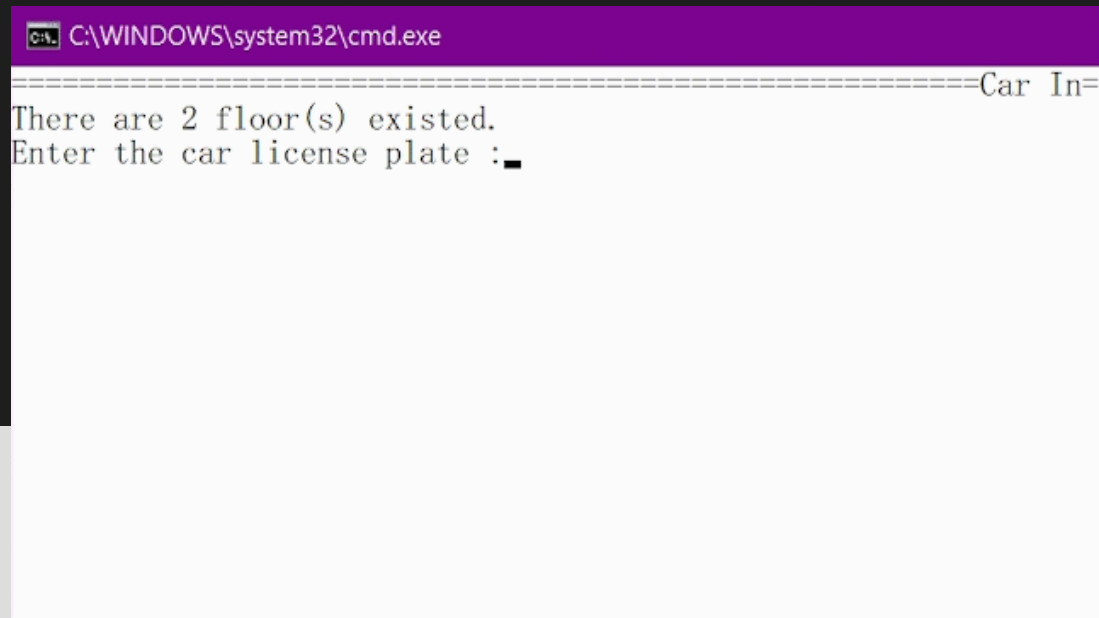
CARIN FUNCTION

```c
int carIn() {
    int address;
    int *time;
    // printf the menu
    system("cls");
    printf("=========================================================Car In=========================================================\n");
    // Check for empty space in array
    for (address = 0; address < 1000; address++) {
        if (customerArray[address][0] == 0) {
            break;
        }

    }
    //printf("Address = %d\n", address);
    printf("There are %d floor(s) existed.\n", parkingLotRead[0]);
    // Take car license plate
    printf("Enter the car license plate :");
    scanf("%s %d",&licensePlate[address] , &customerArray[address][0]);
    // Check for invalid license plate
    while (customerArray[address][0] < 1 || customerArray[address][0]>9999) {
        printf("Invalid Input!!\n");
        printf("Enter the car license plate :");
        scanf("%s %d", &licensePlate[address], &customerArray[address][0]);
    }
```

```
C:\WINDOWS\system32\cmd.exe
=========================================================Car In=
There are 6 floor(s) existed.
Enter the car license plate :
```

CAR IN FUNCTION

```
// Enter the floor that the car going to park
printf("Enter the floor : ");
scanf(" %d", &customerArray[address][13]);
// Check if the input is invalid or the floor is full
while (customerArray[address][13] < 1 || customerArray[address][13] > parkingLotRead[0] || parkingLotLeft[customerArray[address][13]] == 0) {
    if (customerArray[address][13] < 1 || customerArray[address][13] > parkingLotRead[0]) {
        printf("Invalid Input!!\n");
    }
    else if (parkingLotLeft[customerArray[address][13]] == 0) {
        printf("The selected floor is full, please select a new one!!\n");
    }
    printf("Enter the floor : ");
    scanf(" %d", &customerArray[address][13]);
}
// Take out empty space from that floor
parkingLotLeft[customerArray[address][13]]--;
```



CAR IN FUNCTION

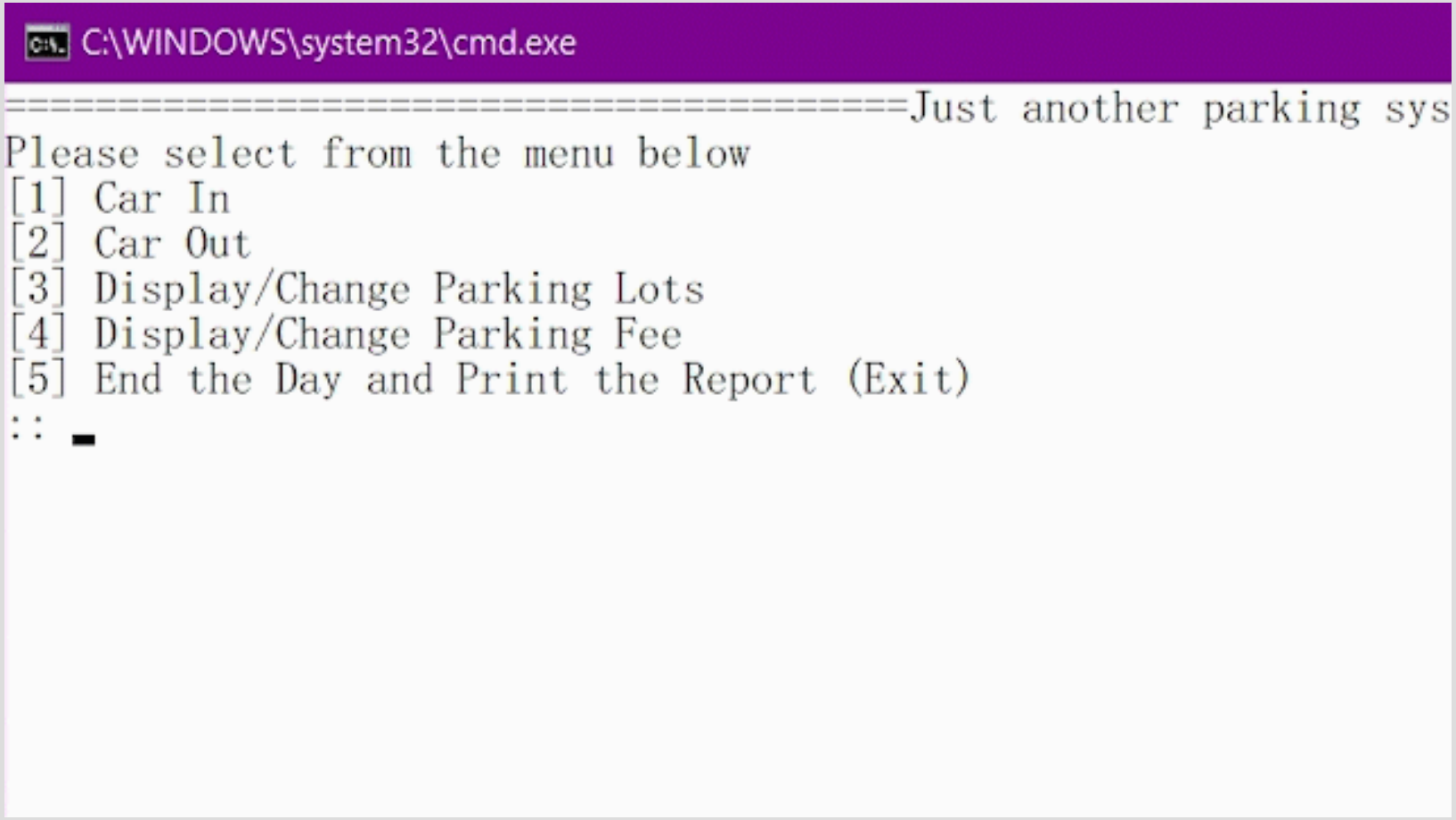
```
// printf the data and save the timestamp to array
printf("The License Plate is %s%d\n", licensePlate[address], customerArray[address][0]);
time = getTime();
customerArray[address][1] = time[0];
customerArray[address][2] = time[1];
customerArray[address][3] = time[2];
customerArray[address][4] = time[3];
customerArray[address][5] = time[4];
customerArray[address][6] = time[5];
printf("Time is %d/%d/%d %02d:%02d:%02d\n", customerArray[address][3], customerArray[address][2], customerArray[address][1], customerArray[address]
printf("Floor parked is %d\n", customerArray[address][13]);
printf("Data Saved!\n");
// Add total car count by one
currentCarCount++;
// Pause the console so user can see the data
system("pause");
// Clear the console to display menu
system("cls");
return 0;
```

```
C:\WINDOWS\system32\cmd.exe
========================================Just another parking sys
Please select from the menu below
[1] Car In
[2] Car Out
[3] Display/Change Parking Lots
[4] Display/Change Parking Fee
[5] End the Day and Print the Report (Exit)
::
```

CAPACITY CHECK FUNCTION

```
int displayParkingLots() {
    // declare variable
    int countPrint = 1;
    int selection = 0;

    // printf the menu and parking space
    while (selection != 3) {
        system("cls");
        printf("=========================================================Parking Lots Left=========================================================\n");
        printf("%-10s %-10s %-10s\n", "Floor", "Capacity", "Left");

        for (countPrint = 1; countPrint <= parkingLotRead[0]; countPrint++) {
            printf("%-10d %-10d %-10d\n", countPrint, parkingLotRead[countPrint], parkingLotLeft[countPrint]);
        }
        printf("============================================================Menu============================================================\n");
        printf("[1] List car in specific floor\n");
        printf("[2] Change the parking space\n");
        printf("[3] Back to main menu\n");
        printf(":: ");
        // accept and validate user's selection
```

CAPACITY CHECK FUNCTION



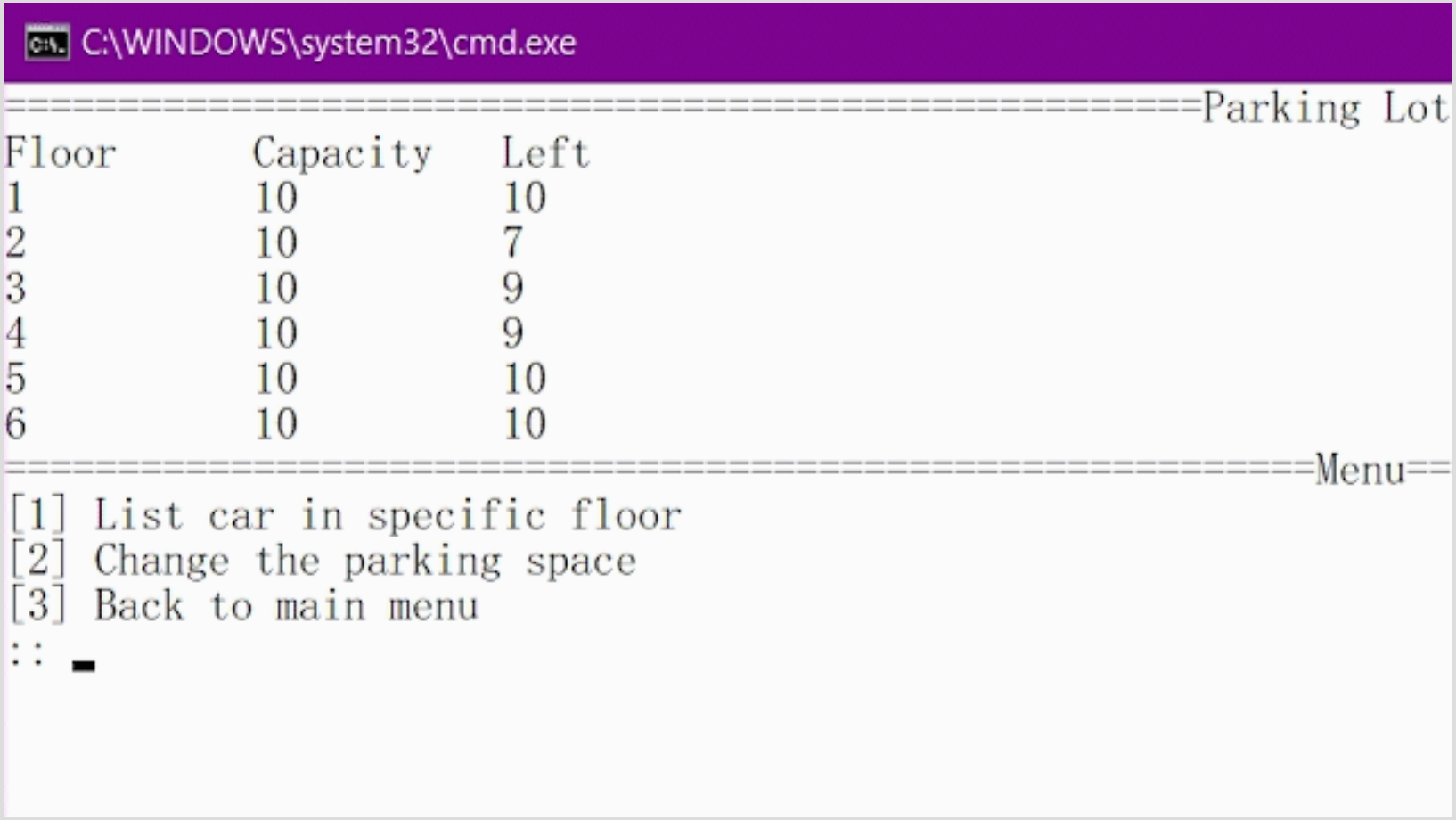
CAPACITY CHECK FUNCTION

```
    // accept and validate user's selection
    scanf("%d", &selection);
    while (selection < 1 || selection > 3) {
        printf("Invalid Selection!!\n");
        printf(":: ");
        scanf("%d", &selection);
    }
    // switch-case based on the selection
    switch (selection) {
    case 1:
        listCar();
        break;
    case 2:
        changeSpace();
        break;
    default:
        break;
    }
}
```

QUERY CAR IN A FLOOR

```
void listCar() {
    int floor,address;
    // Clear previous screen
    system("cls");
    // Receive the floor number input from user
    printf("Enter floor : ");
    scanf(" %d", &floor);
    printf("\n\n==========================================================Car in Floor No. %d==========================================\n", floor);
    // Query all car on that floor
    for (address = 0; address < 1000; address++) {
        if (customerArray[address][7] > 2000) {

        }
        else if (customerArray[address][13] == floor) {
            printf("%c%c%c%d\n",  licensePlate[address][0], licensePlate[address][1], licensePlate[address][2], customerArray[address][0]);
        }

    }
    // Pause the console
    system("pause");
    system("cls");
}
```

QUERY CAR IN A FLOOR



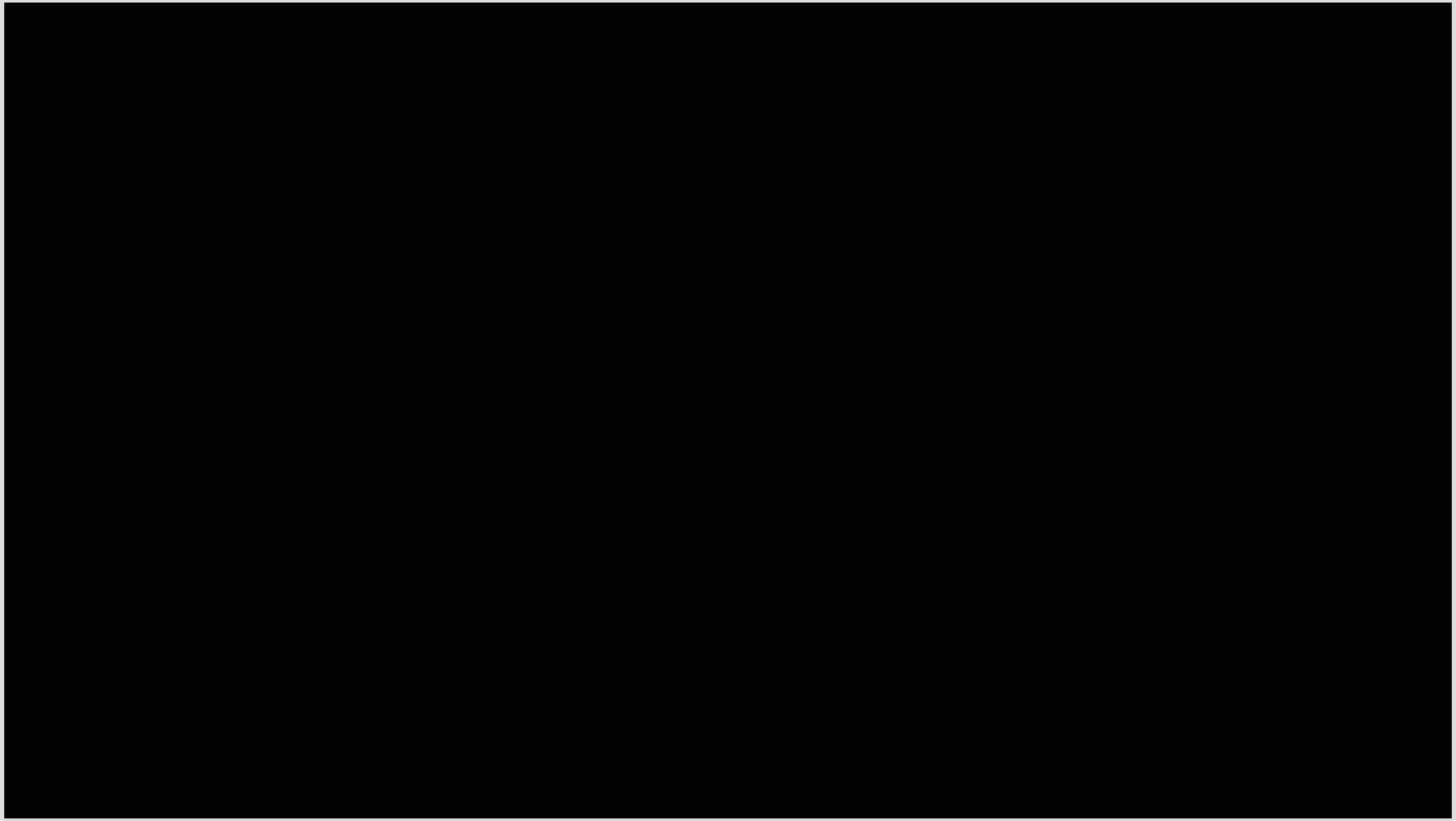
CHANGE PARKING LOT SPACE

```
void changeSpace() {
    // Declare Variables
    char confirm;
    int *parkingLotNumber;
    int countTransfer;
    // Clear screen and print the menu
    system("cls");
    printf("[Warning] All car data will be in this process, continue? (Y/N) : ");
    scanf(" %c", &confirm);
    // If yes, clear the array and launch the setup wizard.
    if (confirm == 'Y') {
        int i, j;
        for (i = 0; i < 1000; i++) {
            for (j = 0; j < 15; j++) {
                customerArray[i][j] = 0;
            }
        }

        customerCount = 0;
        for (i = 0; i < 1000; i++) {
            for (j = 0; j < 3; j++) {
                licensePlate[i][j]='\0';
            }
        }
        for (i = 0; i < 100; i++) {
            parkingLotRead[i]=0;
            parkingLotLeft[i]=0;
        }
```

CHANGE PARKING LOT SPACE

```
parkingLotNumber=parkingLotsNumberSetup();
// Transfer new data to array
for (countTransfer = 0; countTransfer < 100; countTransfer++) {
    parkingLotRead[countTransfer] = parkingLotNumber[countTransfer];
}
for (countTransfer = 0; countTransfer < 100; countTransfer++) {
    parkingLotLeft[countTransfer] = parkingLotRead[countTransfer];
}
// Write the changes to file
FILE *file;
file = fopen("config.txt", "w");
fprintf(file, "Floor=%d\n", parkingLotNumber[0]);
for (count = 1; count <= 99; count++) {
    fprintf(file, "Floor %d=%d\n", count, parkingLotNumber[count]);
}
fprintf(file, "Rate=%d\n", parkingRate[10][0]);
for (countRate = 0; countRate <= 9; countRate++) {
    fprintf(file, "R %d %d\n", parkingRate[countRate][0], parkingRate[countRate][1]);
}
fclose(file);

printf("Floor Updated!\n");
system("pause");
```

CHANGE PARKING LOT SPACE



VIEW/CHANGE PARKING RATE

```
void changeFee() {
    // Declare Variables
    char confirm;
    int i, j, displayLoop, selection=0;
    // Clear the console
    system("cls");
    // printf the parking fee rate
    while (selection != 2) {
        system("cls");
        printf("%-15s %-15s\n", "Time (Minutes)", "Fee per hour");
        for (displayLoop = 0; displayLoop < parkingRate[10][0]; displayLoop++) {
            printf("%-15d %-15d\n", parkingRate[displayLoop][0], parkingRate[displayLoop][1]);
        }
    // printf the menu

        printf("============================================================Menu============================================================\n");
        printf("[1] Change Parking Fee\n");
        printf("[2] Back to main menu\n");
        printf(":: ");
        scanf("%d", &selection);
        // check if the selection valid
        while (selection < 1 || selection > 3) {
            printf("Invalid Selection!!\n");
            printf(":: ");
            scanf("%d", &selection);
```

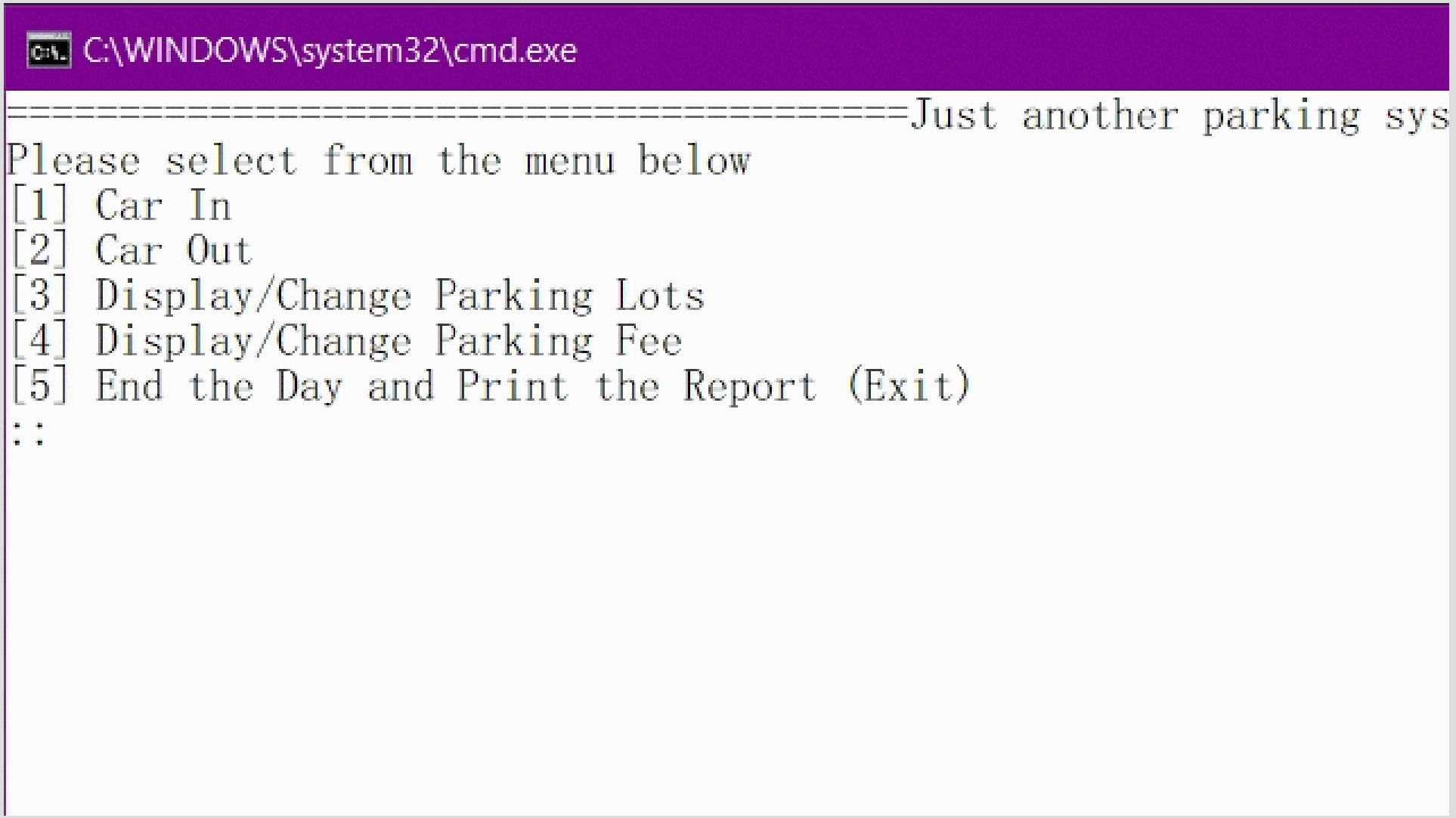
VIEW/CHANGE PARKING RATE

```
        }
        // clear the console
        system("cls");
        switch (selection) {
            //if user select 1, process to parking rate modification wizard.
        case 1:
            //ask for the last confirmation.
            printf("[Warning] All parking rates will be in this process, continue? (Y/N) : ");
            scanf(" %c", &confirm);
            if (confirm == 'Y') {
                // clear the array
                for (i = 0; i < 1000; i++) {
                    for (j = 0; j < 15; j++) {
                        parkingRate[i][j] = 0;
                    }
                }
                // launch the wizard
                parkingFeeRateSetup(parkingRate);
```

VIEW/CHANGE PARKING RATE

```
// write changes to file
FILE *file;
file = fopen("config.txt", "w");
fprintf(file, "Floor=%d\n", parkingLotRead[0]);
for (count = 1; count <= 99; count++) {
    fprintf(file, "Floor %d=%d\n", count, parkingLotRead[count]);
}
fprintf(file, "Rate=%d\n", parkingRate[10][0]);
for (countRate = 0; countRate <= 9; countRate++) {
    fprintf(file, "R %d %d\n", parkingRate[countRate][0], parkingRate[countRate][1]);
}
fclose(file);
// inform the user and pause the console.
printf("Rate Updated!\n");
system("pause");
system("cls");
```

VIEW/CHANGE PARKING RATE



CAR OUT FUNCTION

```
int carOut() {
    // Declare variables
    int plateNumber, address;
    int *time;
    int hrs=0;
    int diffHours, diffMinutes, diffSeconds;
    int countAddLotFree;
    int loop = 0, rateAddress;
    int rateCalCount;
    int totalSecondParked;
    int fee = 0, currentRate;
    char plateLetter[4];
    system("cls");
    // printf the menu
    printf("=========================================================Car Out=============================================================\n");
    // check if car exist in the parking facility
    if (currentCarCount != 0) {
        printf("Enter the car license plate :");
        scanf("%s %d", &plateLetter, &plateNumber);
        // receive the license plate and validates
        for (address = 0; address < 1001; address++) {
            // printf("%d %d", plateNumber, customerArray[address][0]);
            //printf("%c %c %c %c %c %c", plateLetter[0], plateLetter[1], plateLetter[2], licensePlate[address][0], licensePlate[address][1], licensePlate[address][2]
            if (plateLetter[0] == licensePlate[address][0] && plateLetter[1] == licensePlate[address][1] && plateLetter[2] == licensePlate[address][2] && plateNumber
                break;
            }
        }
```

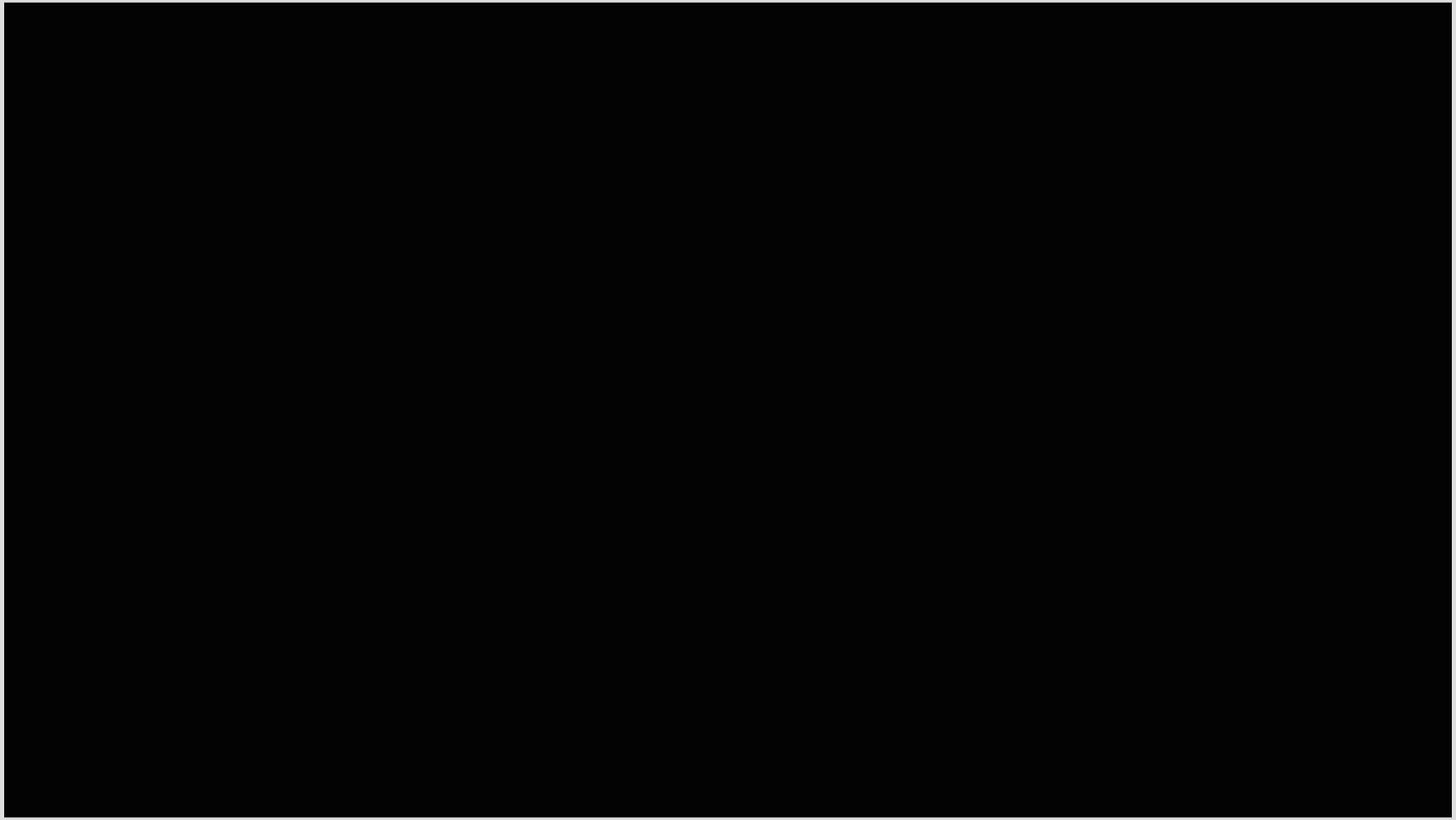
```
// check if the license plate valid
while (address == 1001) {
    printf("Not Found!! Please Re-enter\n");
    printf("Enter the car license plate :");
    scanf("%s %d", &plateLetter, &plateNumber);
    for (address = 0; address < 1001; address++) {
        if (plateLetter[0] == licensePlate[address][0] && plateLetter[1] == licensePlate[address][1] && plateLetter[2] == licensePlate[address][2] &&
            break;
        }
    }
}
// just another debugging stuff
printf("Address = %d\n", address);
// read time from the system and store in car out time
time = getTime();
customerArray[address][7] = time[0];
customerArray[address][8] = time[1];
customerArray[address][9] = time[2];
customerArray[address][10] = time[3];
customerArray[address][11] = time[4];
customerArray[address][12] = time[5];
// find the parking time for the car by subtract final time by initial
diffHours = customerArray[address][10] - customerArray[address][4];
diffMinutes = customerArray[address][11] - customerArray[address][5];
diffSeconds = customerArray[address][12] - customerArray[address][6];
// time digit stuff
if (diffSeconds < 0) {
    diffSeconds = diffSeconds + 60;
    diffMinutes--;
}
if (diffMinutes < 0) {
    diffMinutes = diffMinutes + 60;
    diffHours--;
}
printf("Parked for %02d:%02d:%02d\n", diffHours, diffMinutes, diffSeconds);
```

```
printf("Parked for %02d:%02d:%02d\n", diffHours, diffMinutes, diffSeconds);
//Calculate Parking Fee
totalSecondParked = diffSeconds;
totalSecondParked = totalSecondParked + (diffMinutes * 60);
totalSecondParked = totalSecondParked + (diffHours * 3600);
//Debug
//totalSecondParked = 36001;
printf("Parked for %d second, calculating the fee with %d rate...\n", totalSecondParked, parkingRate[10][0]);
//Adding fee
if (totalSecondParked <= (parkingRate[0][0]*60)) {
    fee = 0;
}
else {
    while (totalSecondParked > 0) {
        for (loop = 1; loop < parkingRate[10][0]; loop++) {
            if (parkingRate[loop][0] > (hrs * 60)) {
                rateAddress = loop;
                currentRate = parkingRate[rateAddress][1];
                break;
            }
        }

        fee = fee + currentRate;
        hrs++;
        totalSecondParked = totalSecondParked - 3600;
    }
}
// print the fee
printf("The Parking fee is %d baht.\n", fee);
parkingLotLeft[customerArray[address][13]]++;
customerArray[address][14] = fee;
currentCarCount--;
```

CAR OUT FUNCTION

```
// If no car is found in the array
else {
    printf("Error! No car parked yet!!\n");
}
// pause the console
system("pause");
system("cls");
return 0;
```

CAR OUT FUNCTION



LOG WRITING FUNCTION

```
void writeLog() {
    // declare variables
    int *time;
    int d, m, y;
    int count = 0;
    int tAddress = 0;
    int address = 0;
    int totalFee = 0;
    int diffHours, diffMinutes, diffSeconds;
    // read date from the system
    time = getTime();
    d = time[2];
    m = time[1];
    y = time[0];
    char str[20];
    //concat the string to make the file name
    sprintf(str, "%02d-%02d-%d", d,m,y);
    strcat(str, "-log.txt");
    printf("File name %s\n", str);
    //open file for writing
    FILE *file;
    file = fopen(str, "w");
    //file content writing starting here
```

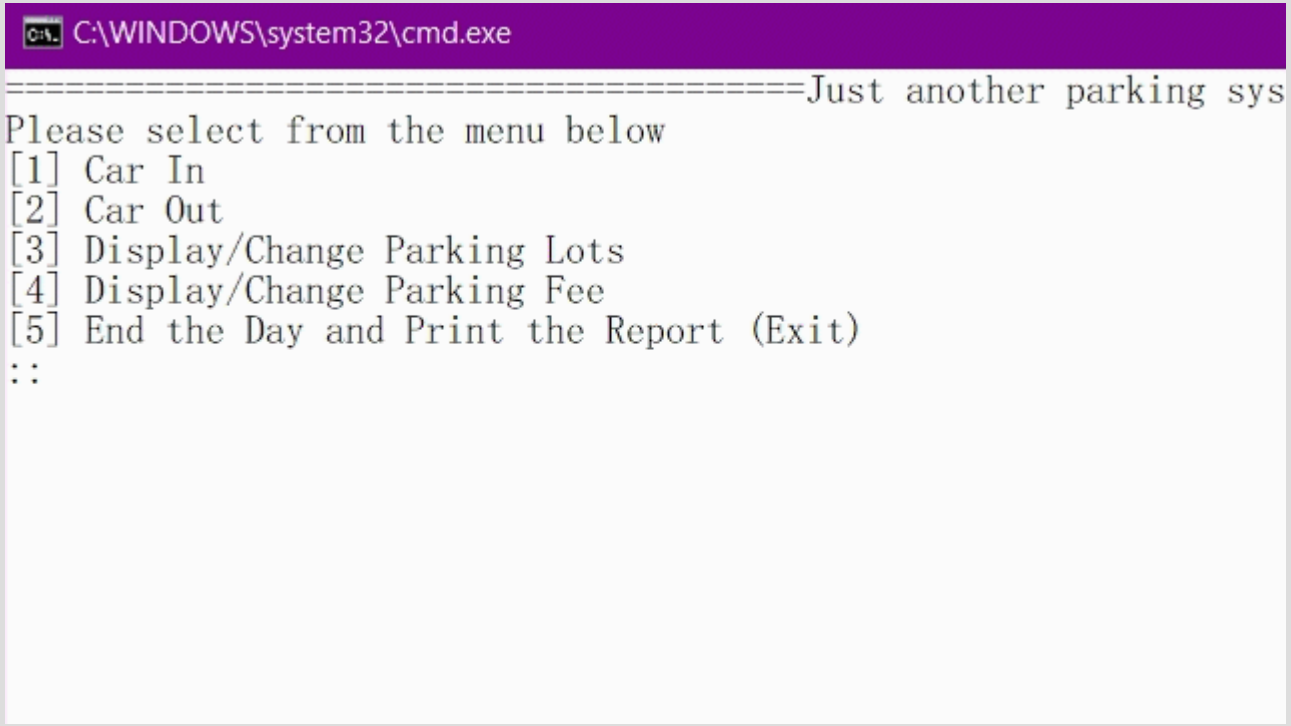
LOG WRITING FUNCTION

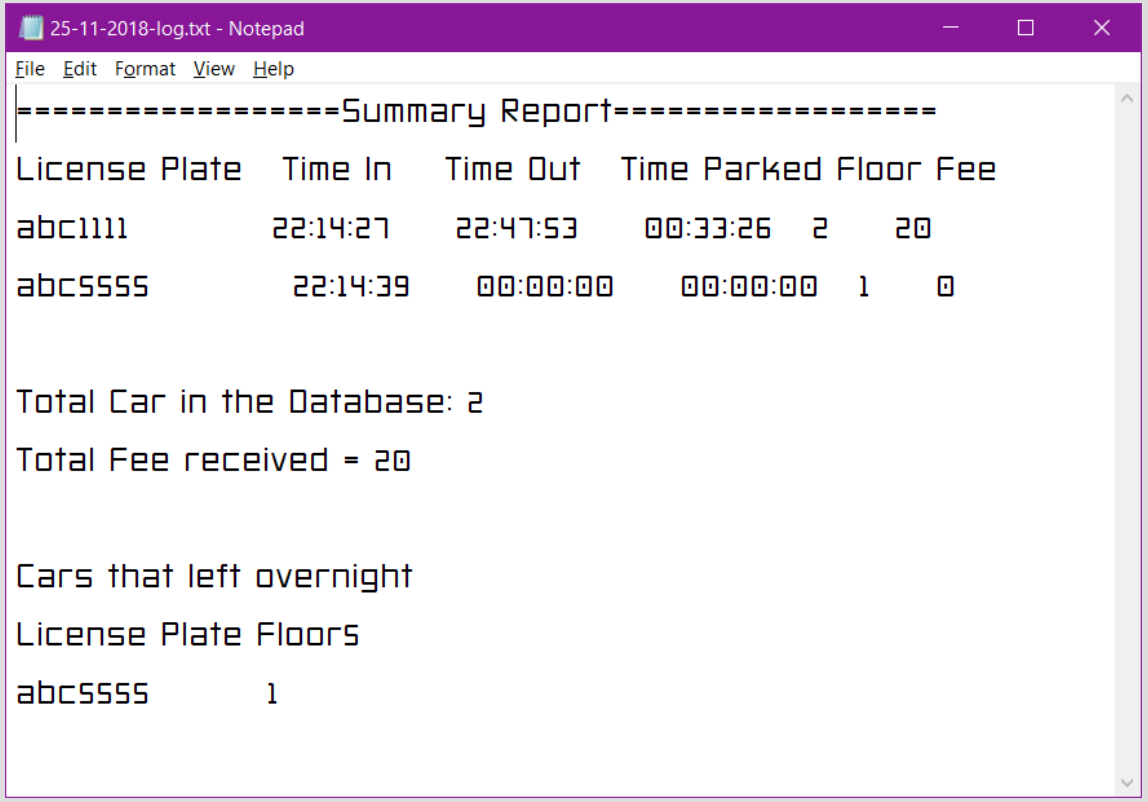
```
//file content writing starting here
fprintf(file, "==================Summary Report==================\n");
fprintf(file, "%-15s %-10s %-10s %-10s %-5s %-4s\n", "License Plate", "Time In", "Time Out", "Time Parked", "Floor", "Fee");
//writing element of an array up to 1,000 of them
for (tAddress = 0; tAddress < 1000; tAddress++) {
    diffHours = customerArray[tAddress][10] - customerArray[tAddress][4];
    diffMinutes = customerArray[tAddress][11] - customerArray[tAddress][5];
    diffSeconds = customerArray[tAddress][12] - customerArray[tAddress][6];
    if (diffSeconds < 0) {
        diffSeconds = diffSeconds + 60;
        diffMinutes--;
    }
    if (diffMinutes < 0) {
        diffMinutes = diffMinutes + 60;
        diffHours--;
    }
    if (diffHours < 0) {
        diffHours = diffMinutes = diffSeconds = 0;
    }
    fprintf(file, "%c%c%c%-12d   %02d:%02d:%02d     %02d:%02d:%02d     %02d:%02d:%02d   %-5d %-4d\n", licensePlate[tAddress][0], licensePlate[tAddress][1],
        customerArray[tAddress][5], customerArray[tAddress][6], customerArray[tAddress][10], customerArray[tAddress][11], customerArray[tAddress][12],
        diffHours,diffMinutes,diffSeconds, customerArray[tAddress][13], customerArray[tAddress][14]);
    totalFee = totalFee + customerArray[tAddress][14];
    //check if the next array element empty
    if (customerArray[tAddress+1][0] == 0) {
        break;
    }
```

LOG WRITING FUNCTION

```
    }
    //car that left overnight
    fprintf(file, "\nTotal Car in the Database: %d\n", tAddress+1);
    fprintf(file, "Total Fee received = %d\n\n", totalFee);
    fprintf(file, "Cars that left overnight\n");
    fprintf(file, "%12s %s5\n", "License Plate", "Floor");
    for (address = 0; address <= tAddress; address++) {
        if (customerArray[address][7] < 2000) {
            fprintf(file,"%c%c%c%-12d %-5d\n", licensePlate[address][0], licensePlate[address][1], licensePlate[address][2], customerArray[address][0],
        }

    }

    //close the file --> save to disk


    fclose(file);
}
```

LOG WRITING FUNCTION





LIVE DEMO

// Actually just a video :P

Debug x86 Local Windows Debugger

ngSystem.c

rkingSystem (Global Scope) changeFee()

```
        // clear the console
        system("cls");
        switch (selection) {
            //if user select 1, process to parking rate modification wizard.
        case 1:
            //ask for the last confirmation.
            printf("[Warning] All parking rates and cars will be in this process, continue? (Y/N) : ");
            scanf(" %c", &confirm);
            if (confirm == 'Y') {
                // clear the array
                for (i = 0; i < 1000; i++) {
                    for (j = 0; j < 15; j++) {
                        parkingRate[i][j] = 0;
                    }
                }
                for (i = 0; i < 1000; i++) {
                    for (j = 0; j < 15; j++) {
                        customerArray[i][j] = 0;
                    }
                }

                customerCount = 0;
                for (i = 0; i < 1000; i++) {
                    for (j = 0; j < 3; j++) {
                        licensePlate[i][j] = '\0';
                    }
                }
```

Kasidis Arunruangsirilert, Less than 5 minutes ago | 1 author, 26 changes

THANK YOU FOR LISTENING

http://bit.ly/ABACComPro

https://github.com/KenSoft/BG1221ComputerProgrammingForEngineering