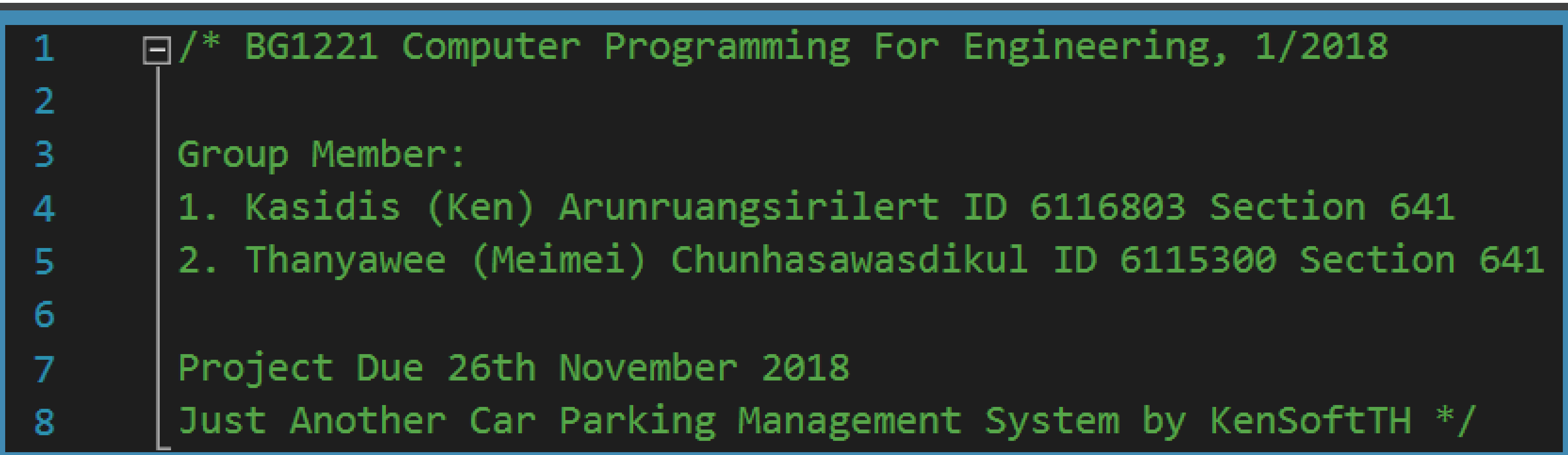
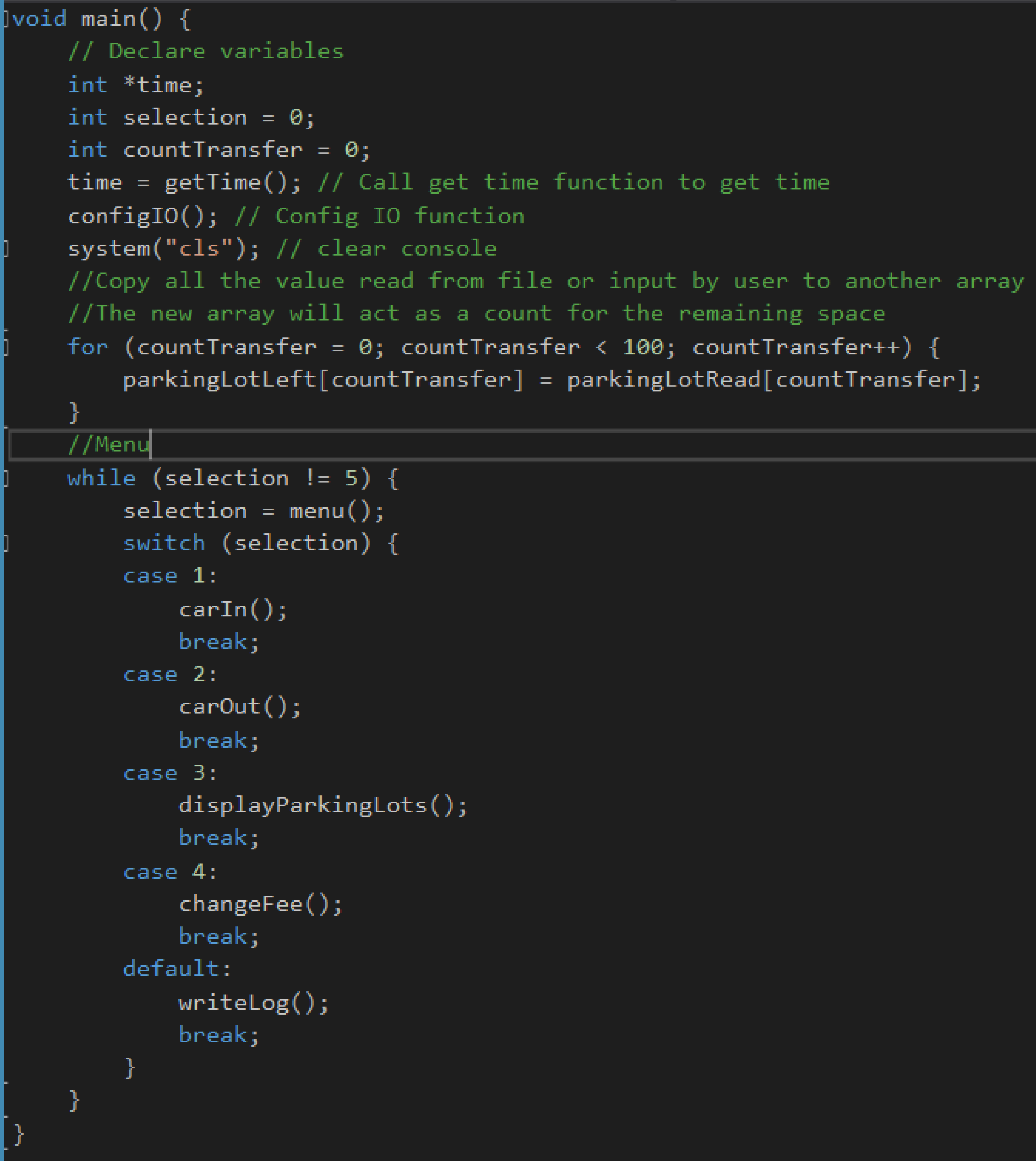
CAR PARKING SYSTEM



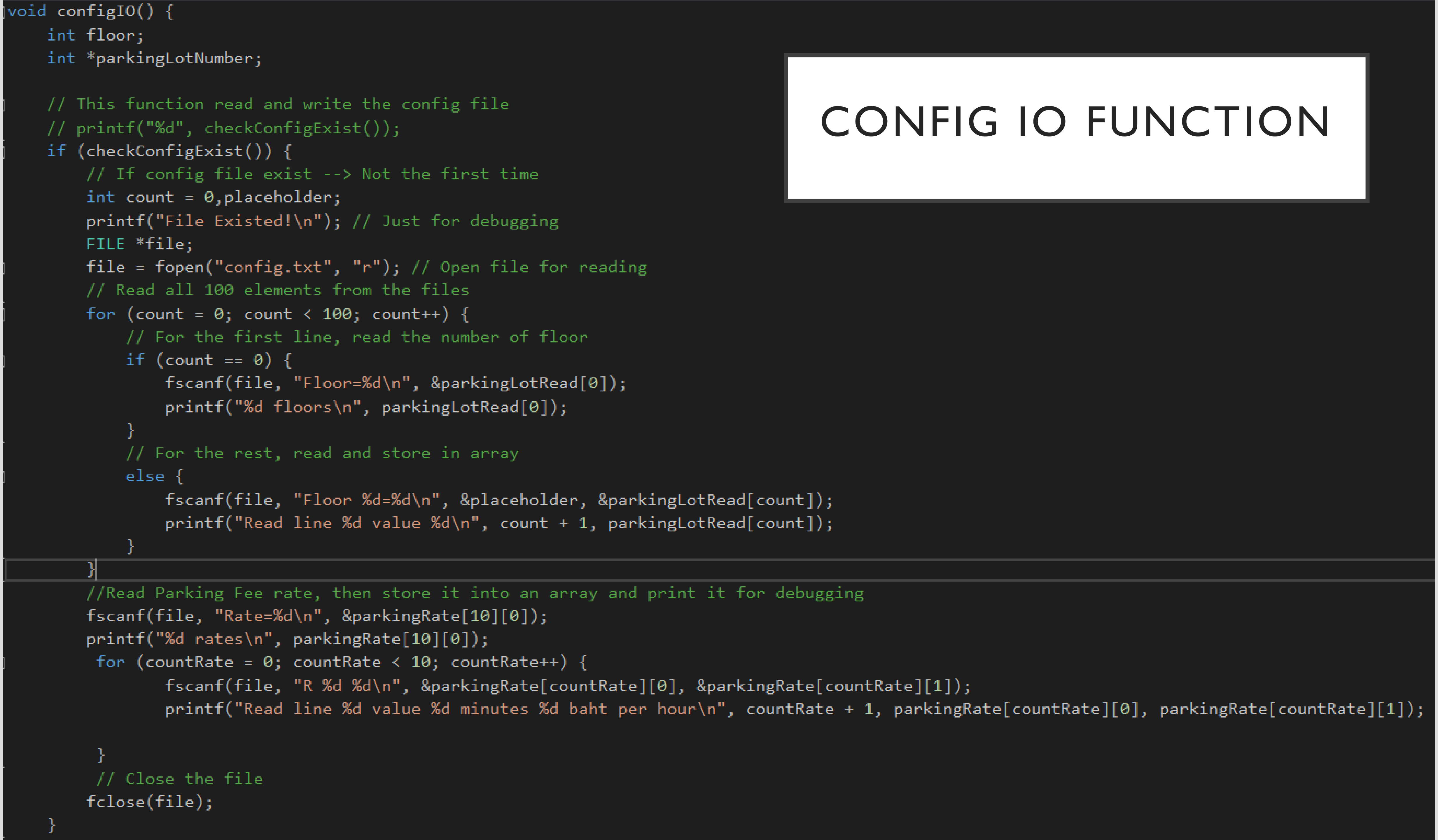
```
/* BG1221 Computer Programming For Engineering, 1/2018

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 */
```

MAIN FUNCTION

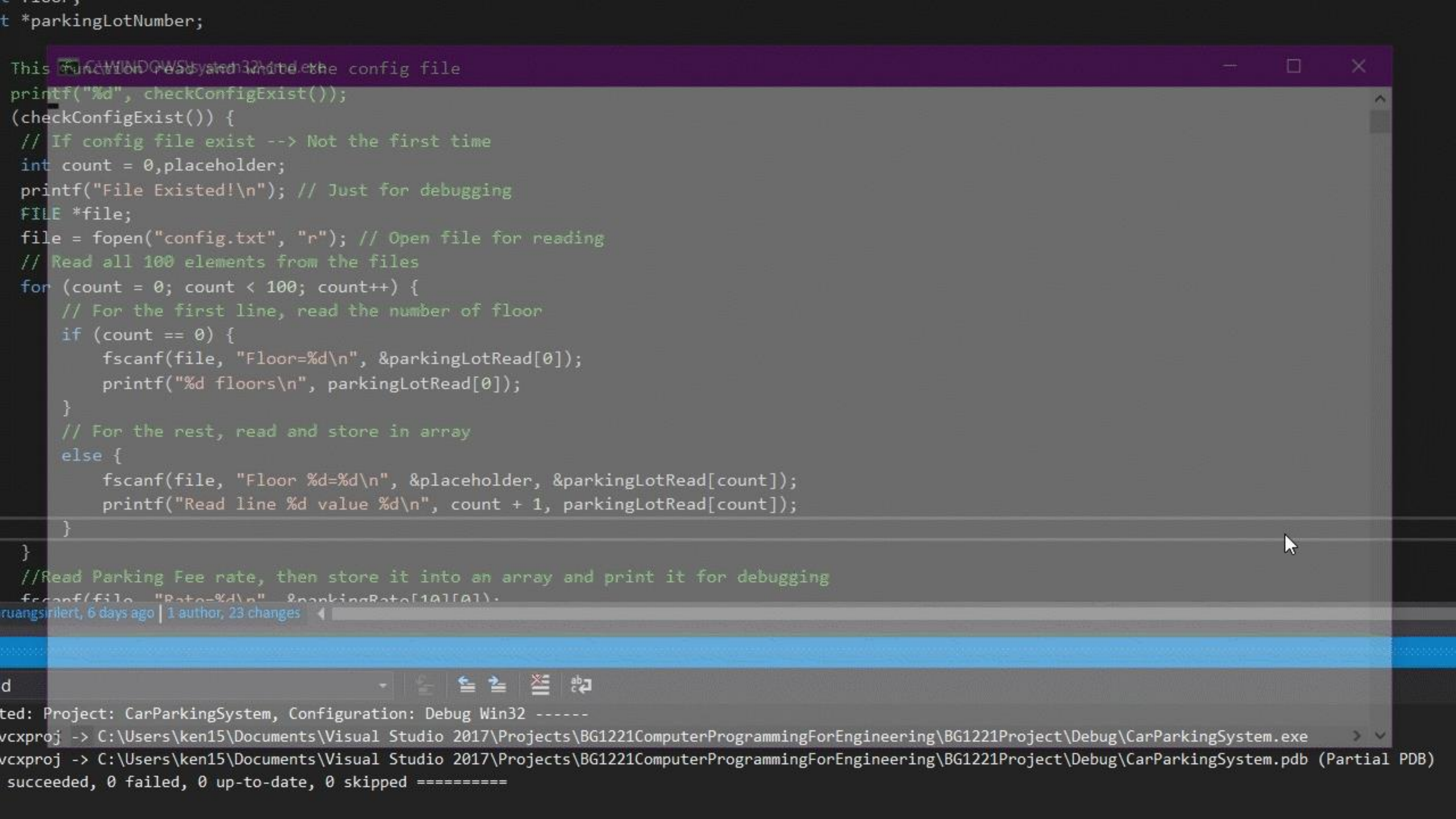


```c
void main() {
    // Declare variables
    int *time;
    int selection = 0;
    int countTransfer = 0;
    time = getTime(); // Call get time function to get time
    configIO(); // Config IO function
    system("cls"); // clear console
    //Copy all the value read from file or input by user to 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);
    }
```

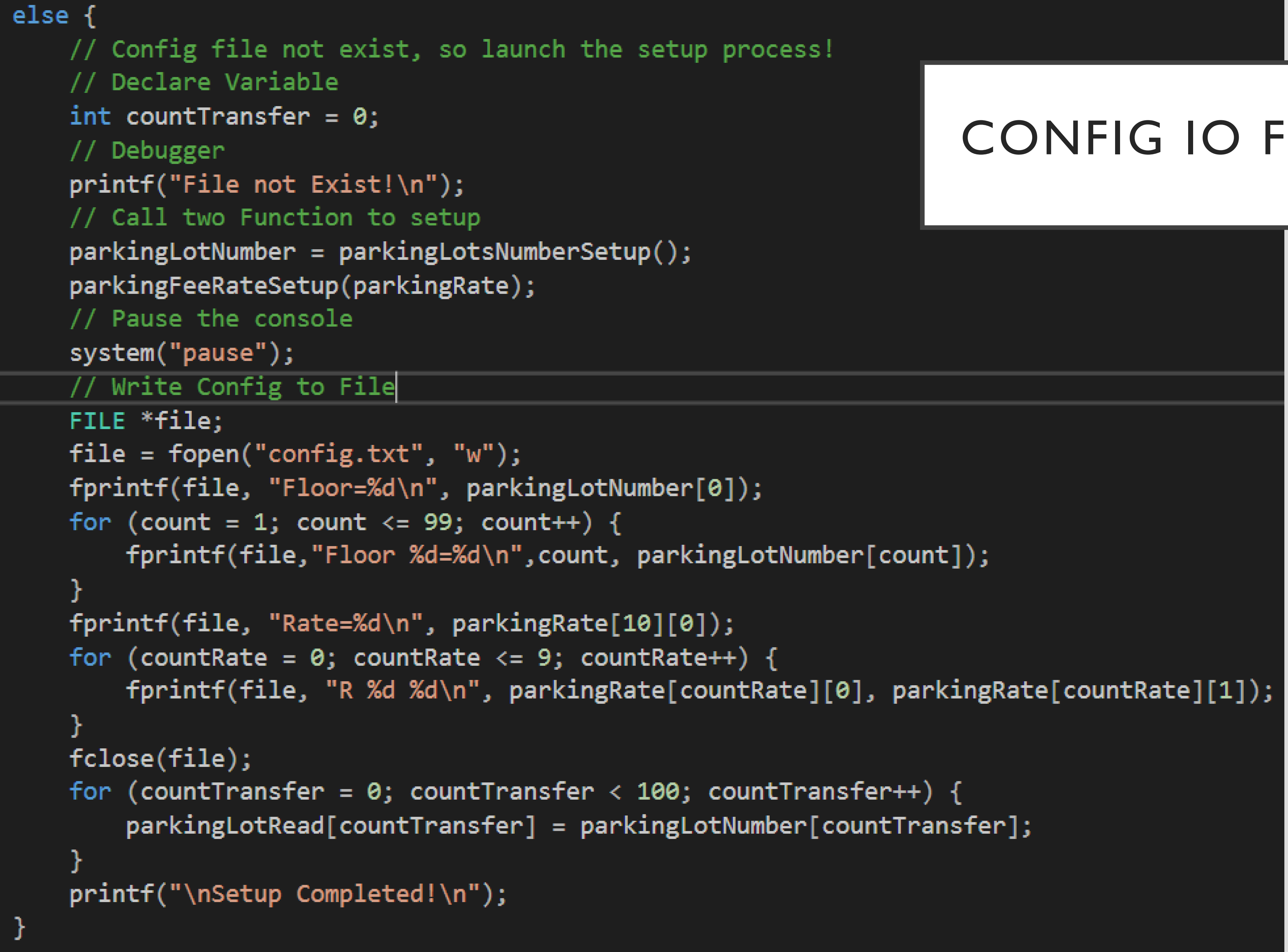


```
t *parkingLotNumber;

 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

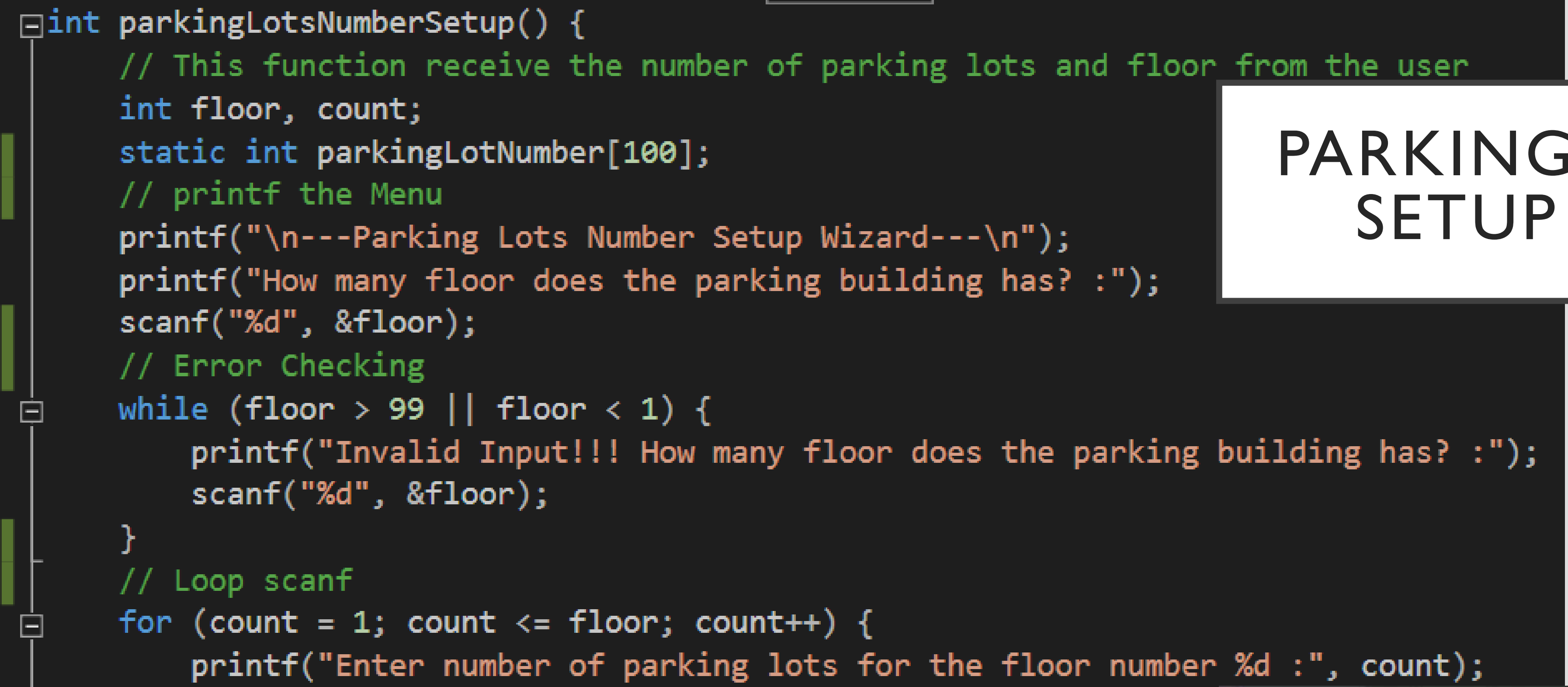
```
ted: Project: CarParkingSystem, Configuration: Debug Win32 ------
vcxproj -> C:\Users\ken15\Documents\Visual Studio 2017\Projects\BG1221ComputerProgrammingForEngineering\BG1221Project\Debug\CarParkingSystem.exe
vcxproj -> C:\Users\ken15\Documents\Visual Studio 2017\Projects\BG1221ComputerProgrammingForEngineering\BG1221Project\Debug\CarParkingSystem.pdb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

CONFIG IO FUNCTION

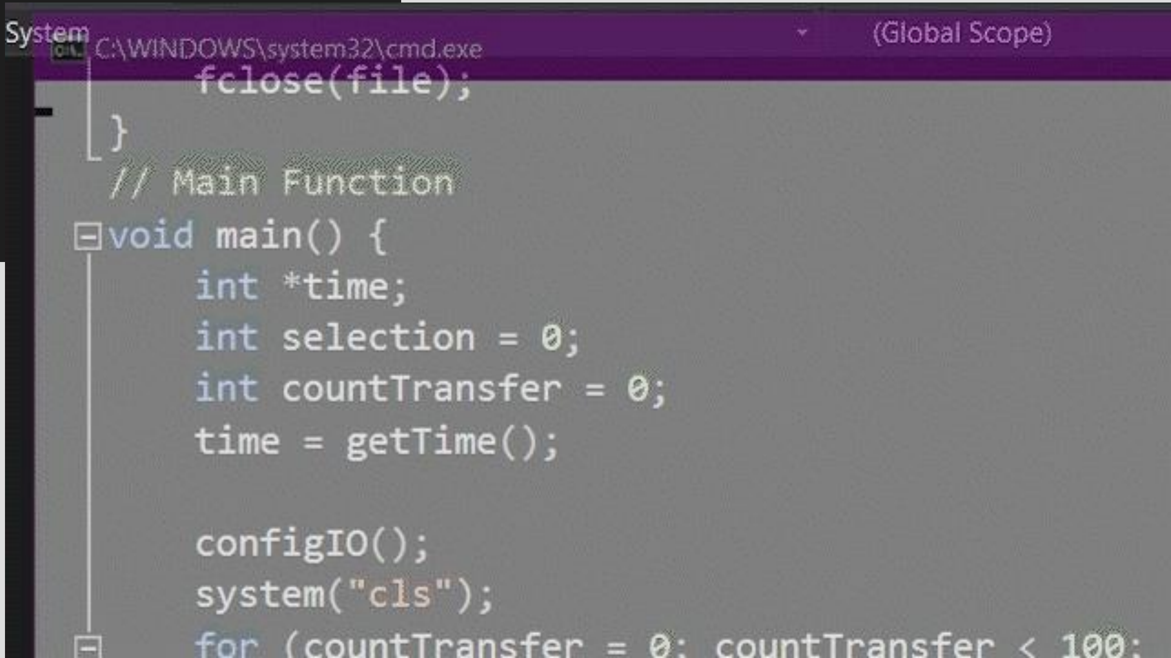


```
else {
    // Config file not exist, so launch the setup process!
    // Declare Variable
    int countTransfer = 0;
    // Debugger
    printf("File not Exist!\n");
    // Call two Function to setup
    parkingLotNumber = parkingLotsNumberSetup();
    parkingFeeRateSetup(parkingRate);
    // Pause 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");
}
```

PARKINGLOTSNUMBER SETUP FUNCTION

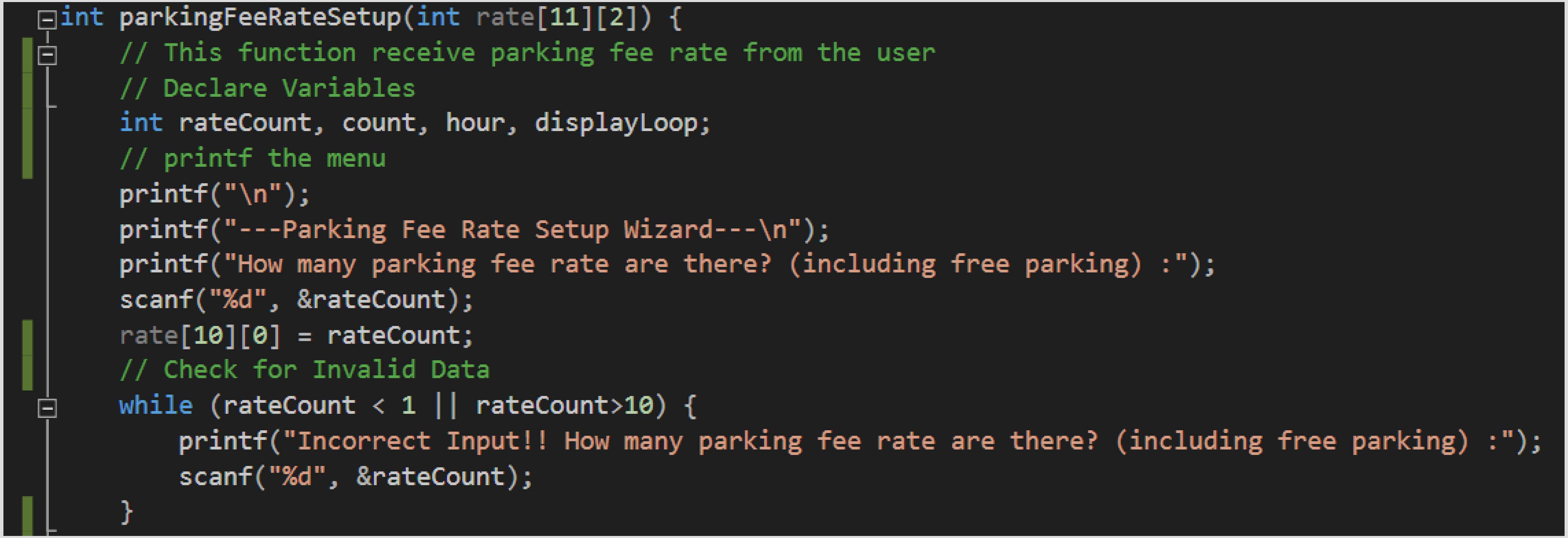


```
int parkingLotsNumberSetup() {
    // This function receive the number of parking lots and floor from the user
    int floor, count;
    static int parkingLotNumber[100];
    // printf the Menu
    printf("\n---Parking Lots Number Setup Wizard---\n");
    printf("How many floor does the parking building 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;
```

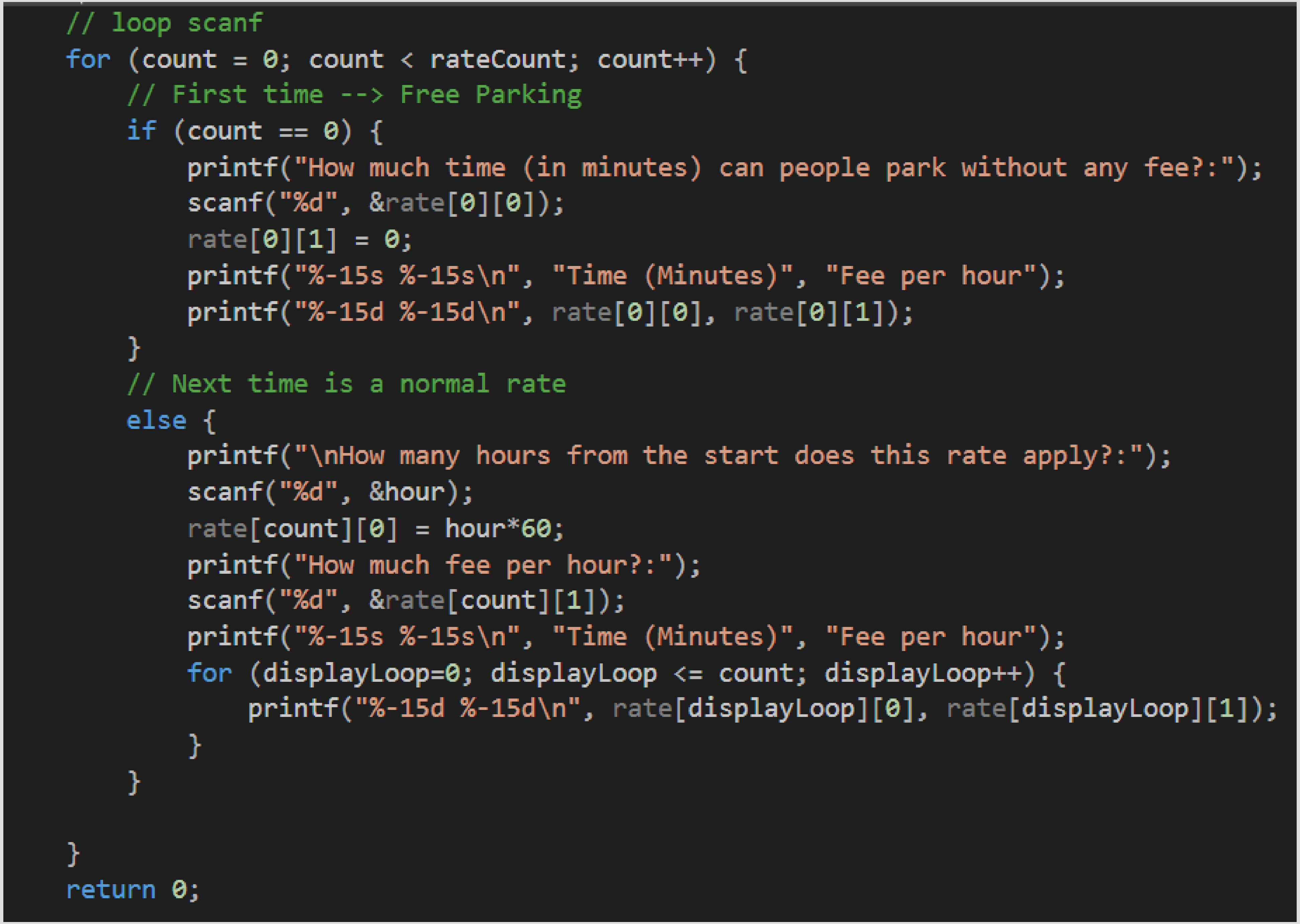


```
        fclose(file);
    }
    // Main Function
void main() {
        int *time;
        int selection = 0;
        int countTransfer = 0;
        time = getTime();

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

PARKINGFEERATESETUP FUNCTION

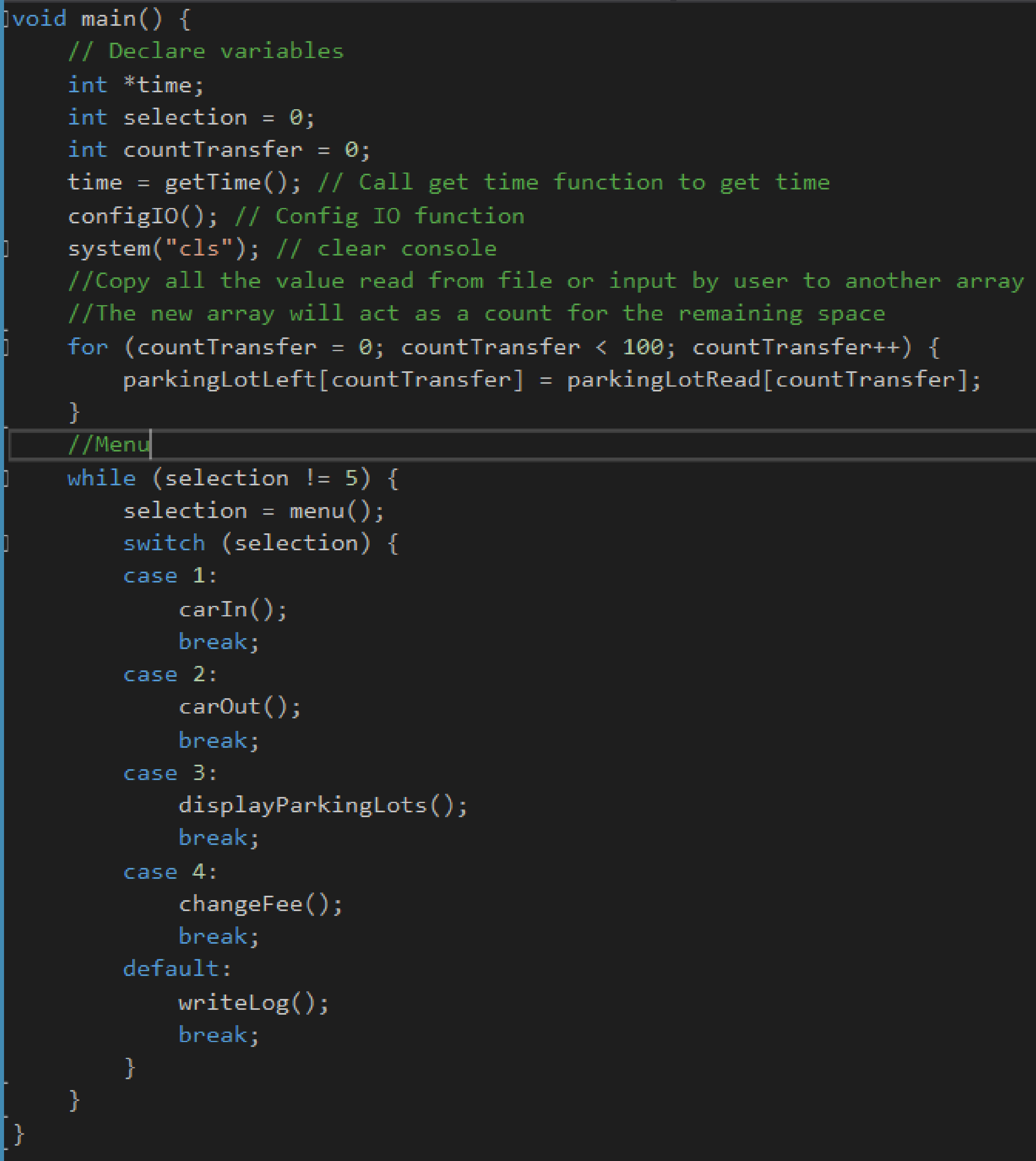


```c
int parkingFeeRateSetup(int rate[11][2]) {
    // This function receive parking fee rate from the user
    // Declare Variables
    int rateCount, count, hour, displayLoop;
    // printf the menu
    printf("\n");
    printf("---Parking Fee Rate Setup Wizard---\n");
    printf("How many parking 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);
    }
```



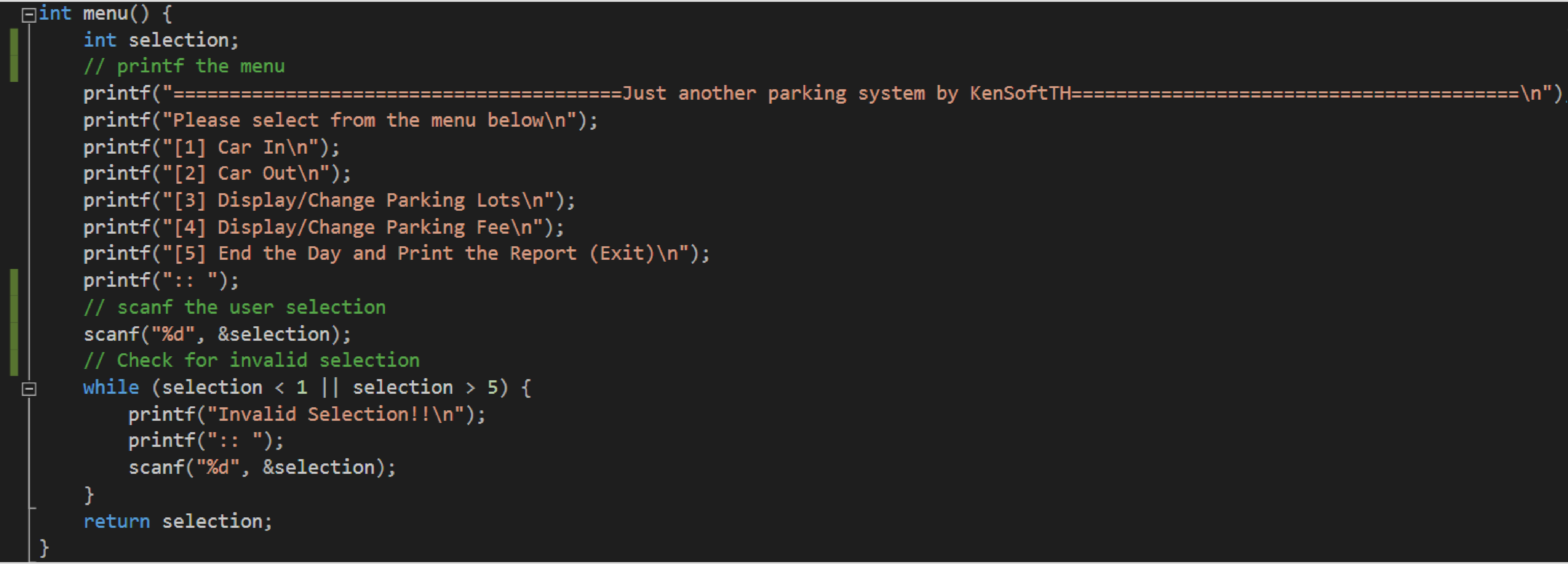
```c
    // loop scanf
    for (count = 0; count < rateCount; count++) {
        // First time --> Free Parking
        if (count == 0) {
            printf("How much time (in minutes) can people park without any fee?:");
            scanf("%d", &rate[0][0]);
            rate[0][1] = 0;
            printf("%-15s %-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;
```

MAIN FUNCTION



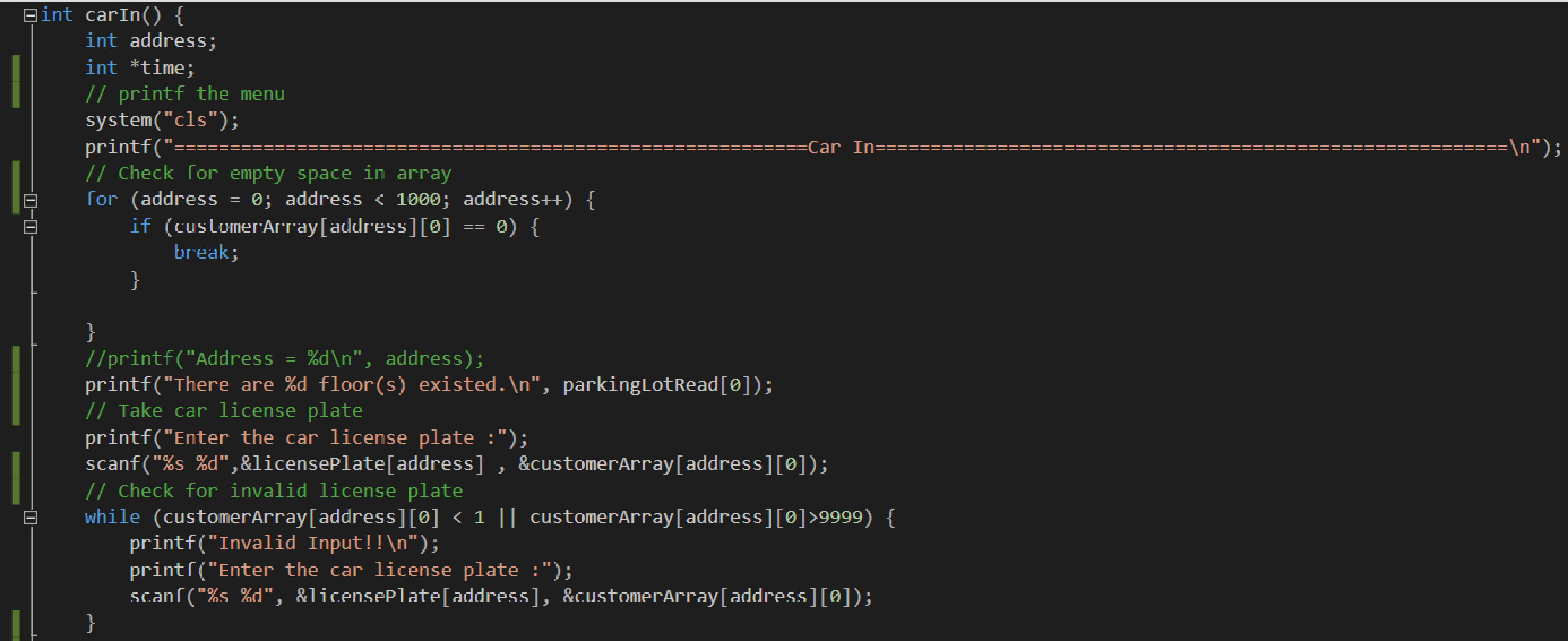
```
void main() {
    // Declare variables
    int *time;
    int selection = 0;
    int countTransfer = 0;
    time = getTime(); // Call get time function to get time
    configIO(); // Config IO function
    system("cls"); // clear console
    //Copy all the value read from file or input by user to 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



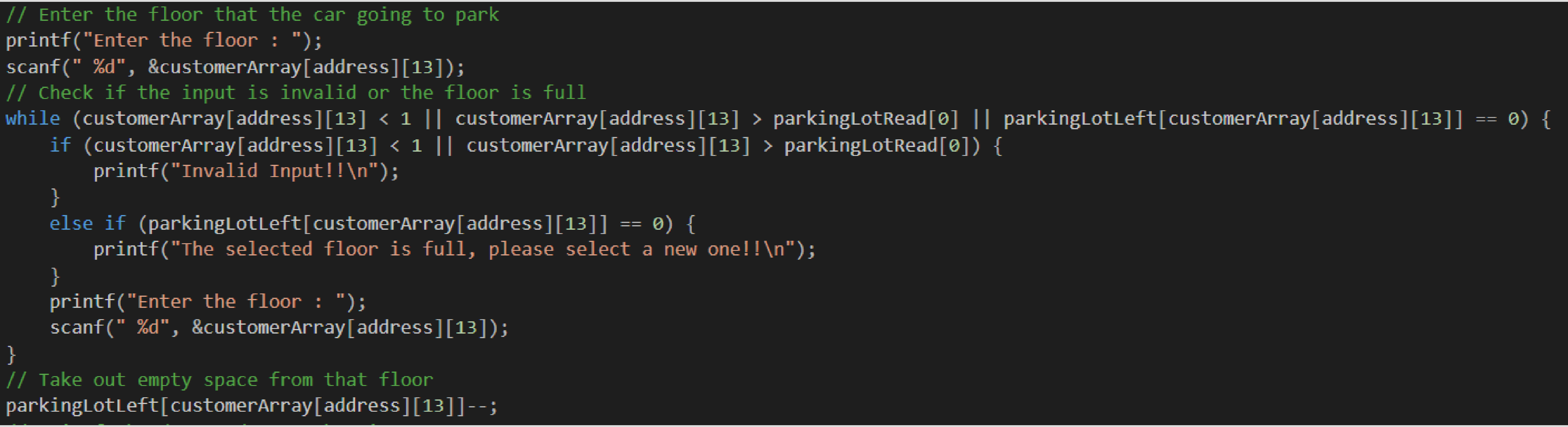
```
int menu() {
    int selection;
    // printf the menu
    printf("========================================Just another parking system by KenSoftTH========================================\n")
    printf("Please select from the menu below\n");
    printf("[1] Car In\n");
    printf("[2] Car Out\n");
    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;
}
```

CARIN FUNCTION



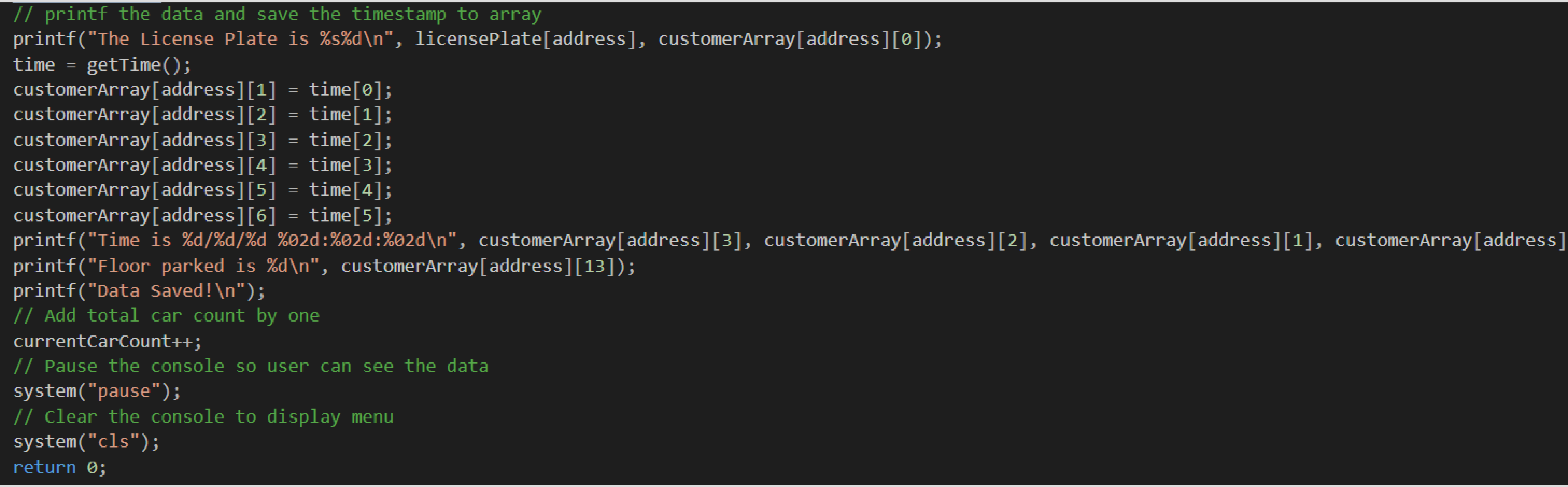
```c
int carIn() {
    int address;
    int *time;
    // printf the menu
    system("cls");
    printf("=========================================================Car In=========================================================\n");
    // Check for empty space in array
    for (address = 0; address <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

CARIN 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]]--;
```

CARIN 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;
```