MOCK TEST

1. Write code for the following:

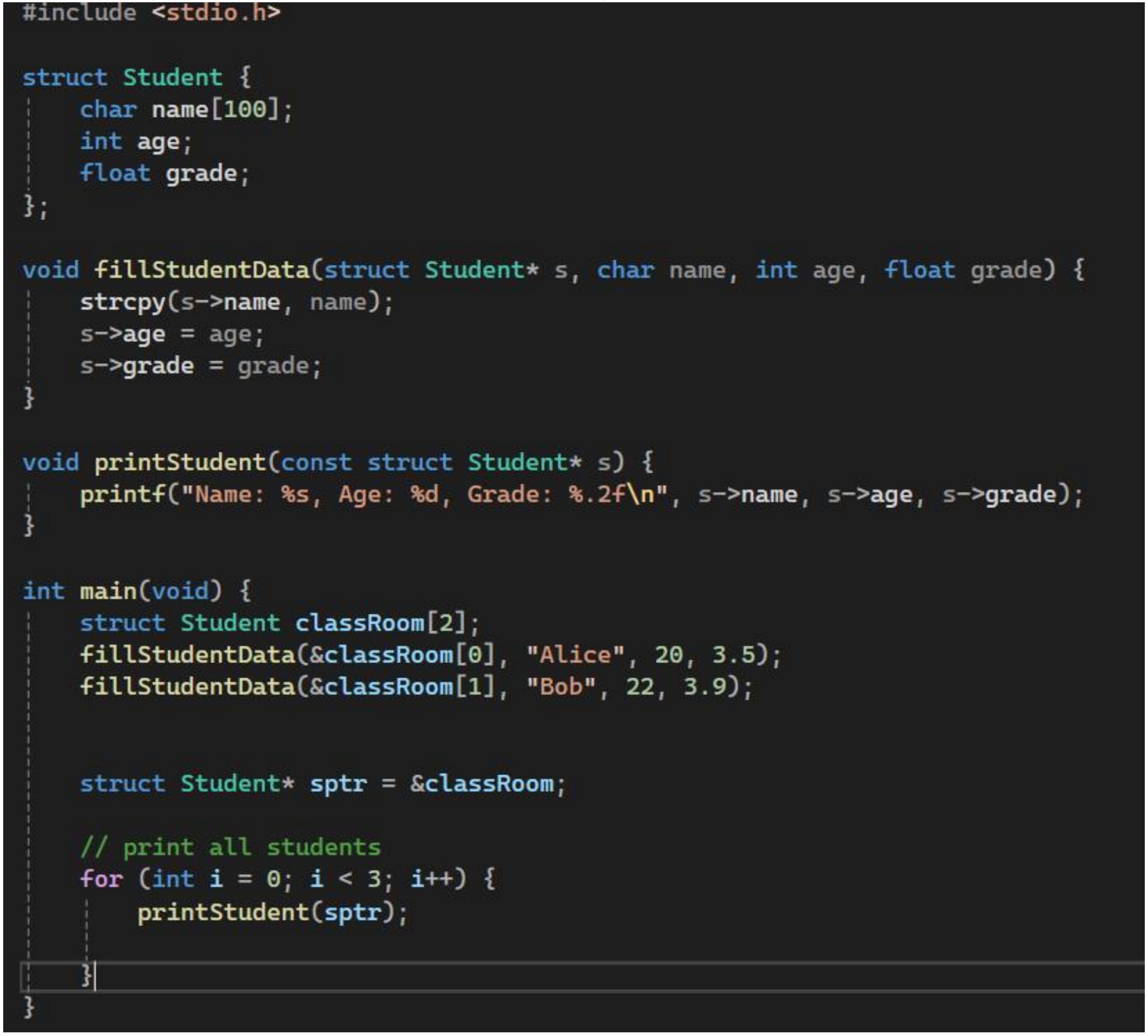
a) Create a Structure called Car. It holds its model year, name, and price.

b) Create a function called viewDetails, which takes a struct Car as an argument and prints out the details in the following way (xxxx are the details):

“Car Name : xxxxxx“
“ Year Made: xxxxx “
“Car Price: xxxxx“

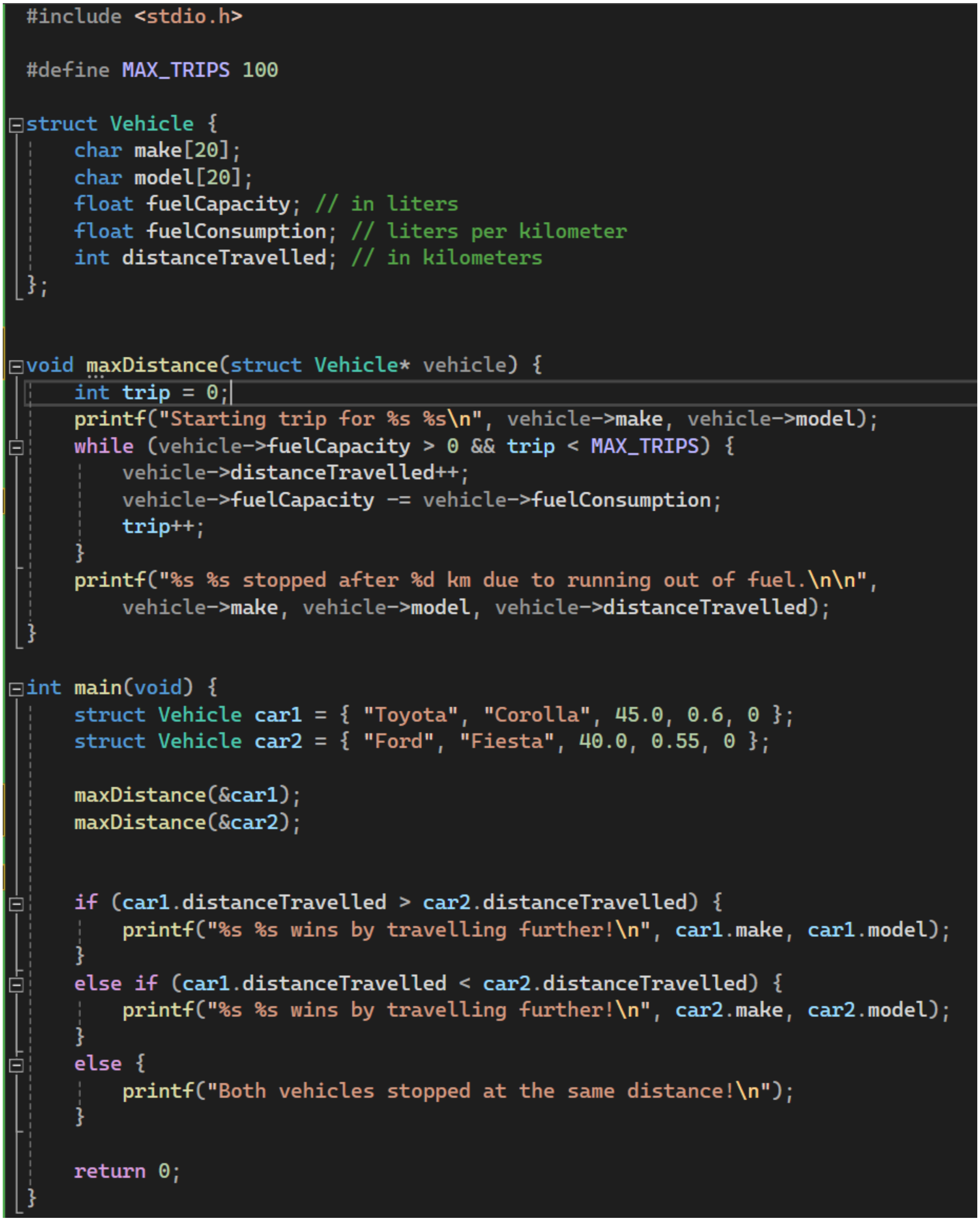
c) In the main function create a struct of Car and call the above function.

2. The following code has some issues. Please debug all of them



```
#include <stdio.h>

struct Student {
    char name[100];
    int age;
    float grade;
};

void fillStudentData(struct Student* s, char name, int age, float grade) {
    strcpy(s->name, name);
    s->age = age;
    s->grade = grade;
}

void printStudent(const struct Student* s) {
    printf("Name: %s, Age: %d, Grade: %.2f\n", s->name, s->age, s->grade);
}

int main(void) {
    struct Student classRoom[2];
    fillStudentData(&classRoom[0], "Alice", 20, 3.5);
    fillStudentData(&classRoom[1], "Bob", 22, 3.9);


    struct Student* sptr = &classRoom;

    // print all students
    for (int i = 0; i < 3; i++) {
        printStudent(sptr);

    }
}
```

3. Write a walkthrough of this code:



```c
#include <stdio.h>

#define MAX_TRIPS 100

struct Vehicle {
    char make[20];
    char model[20];
    float fuelCapacity; // in liters
    float fuelConsumption; // liters per kilometer
    int distanceTravelled; // in kilometers
};


void maxDistance(struct Vehicle* vehicle) {
    int trip = 0;
    printf("Starting trip for %s %s\n", vehicle->make, vehicle->model);
    while (vehicle->fuelCapacity > 0 && trip < MAX_TRIPS) {
        vehicle->distanceTravelled++;
        vehicle->fuelCapacity -= vehicle->fuelConsumption;
        trip++;
    }
    printf("%s %s stopped after %d km due to running out of fuel.\n\n",
        vehicle->make, vehicle->model, vehicle->distanceTravelled);
}

int main(void) {
    struct Vehicle car1 = { "Toyota", "Corolla", 45.0, 0.6, 0 };
    struct Vehicle car2 = { "Ford", "Fiesta", 40.0, 0.55, 0 };

    maxDistance(&car1);
    maxDistance(&car2);


    if (car1.distanceTravelled > car2.distanceTravelled) {
        printf("%s %s wins by travelling further!\n", car1.make, car1.model);
    }
    else if (car1.distanceTravelled < car2.distanceTravelled) {
        printf("%s %s wins by travelling further!\n", car2.make, car2.model);
    }
    else {
        printf("Both vehicles stopped at the same distance!\n");
    }

    return 0;
}
```