a.

The notable obstacles I overcame is when I needed to use “if statement”, I was first struggled where to put “{}”and I often forgot to put “;” after sentences without reminding.

b.

1)(201,1020,10,Shulin,n,12) In this case, the program works well and meets the spec.

2)(201,1020,10,Shulin,n,13) In this case, the program allows me to enter all above inputs, and the result is

“The month number must be in the range 1 through 12.” And meets the spec requirement.

3)

(1212,202,10,Shulin,n,12) In this case, the ending odometer reading is less than the starting reading. If the program is correct, the program will end with a sentence “The ending odometer reading must be at least as large as the starting reading.” and will not allow me to enter the rest input, but if I did not write the “if statement” (**if** (numberAtEnd < numberAtStart) blablabla),the result would be a negative rental charge.

4)

(-1,202,10,Shulin,y,12) In this case, the starting odometer reading is negative. If the program is correct, the program will end with a sentence “The starting odometer reading must not be negative.” and will not allow me to enter the rest input, but if I did not write “return 1” after the “if statement” (**if** (numberAtStart < 0) blablabla), the program can continue processing.

5)

(a,202,10,Shulin,y,12) In this case, the starting odometer reading is not a number but a string. The output is weird as “Odometer at end: Rental days: The number of rental days must be positive.” which is not included in the output the spec gives. We do not need to consider this situation.