Exercise 3.1: Problem 1:

1. Tag 1 intializes the variable int\_var as an interger. Tag 2 prints out a line that says: int data type is (size of (int\_var)) bytes or ( size of (int\_var)\* 8 --- which converts bytes into bits) bits long. Tag 2 pretty much just output the amount of bytes and bits the variable int\_var is.

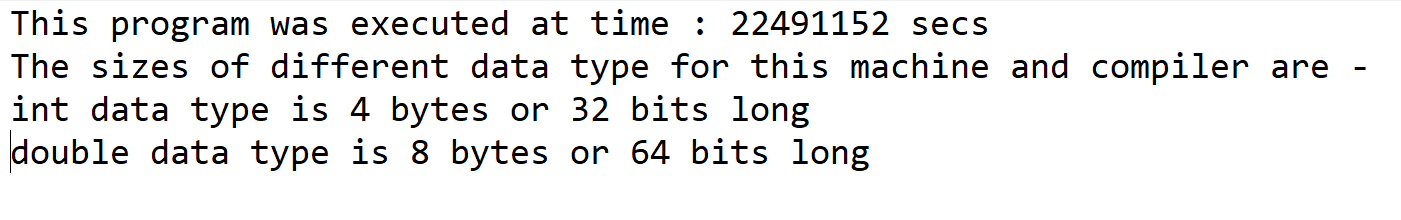
d. There was no negative time. If the was a negative time it would’ve been because of data overflow.

e. There is no notable difference

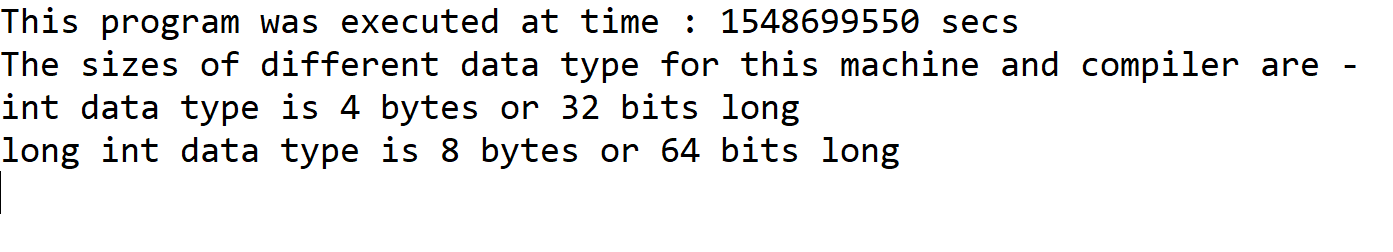
f. Timeval is a struct that represents elapsed time. Each server used has a different timeval because timeval is how long the server has been running

g.

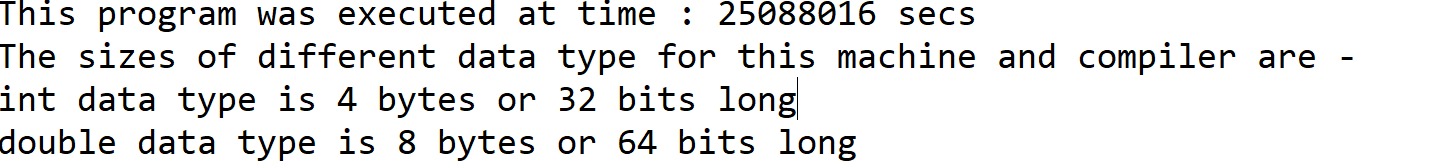
COMPUTE (with double)



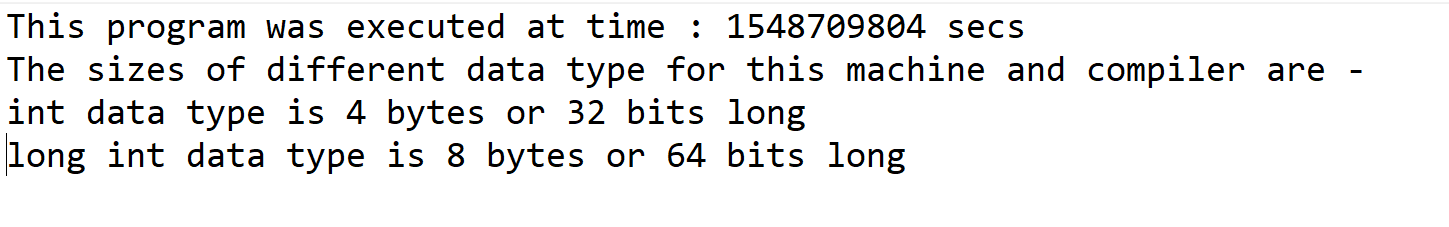
COMPUTE (with long int)



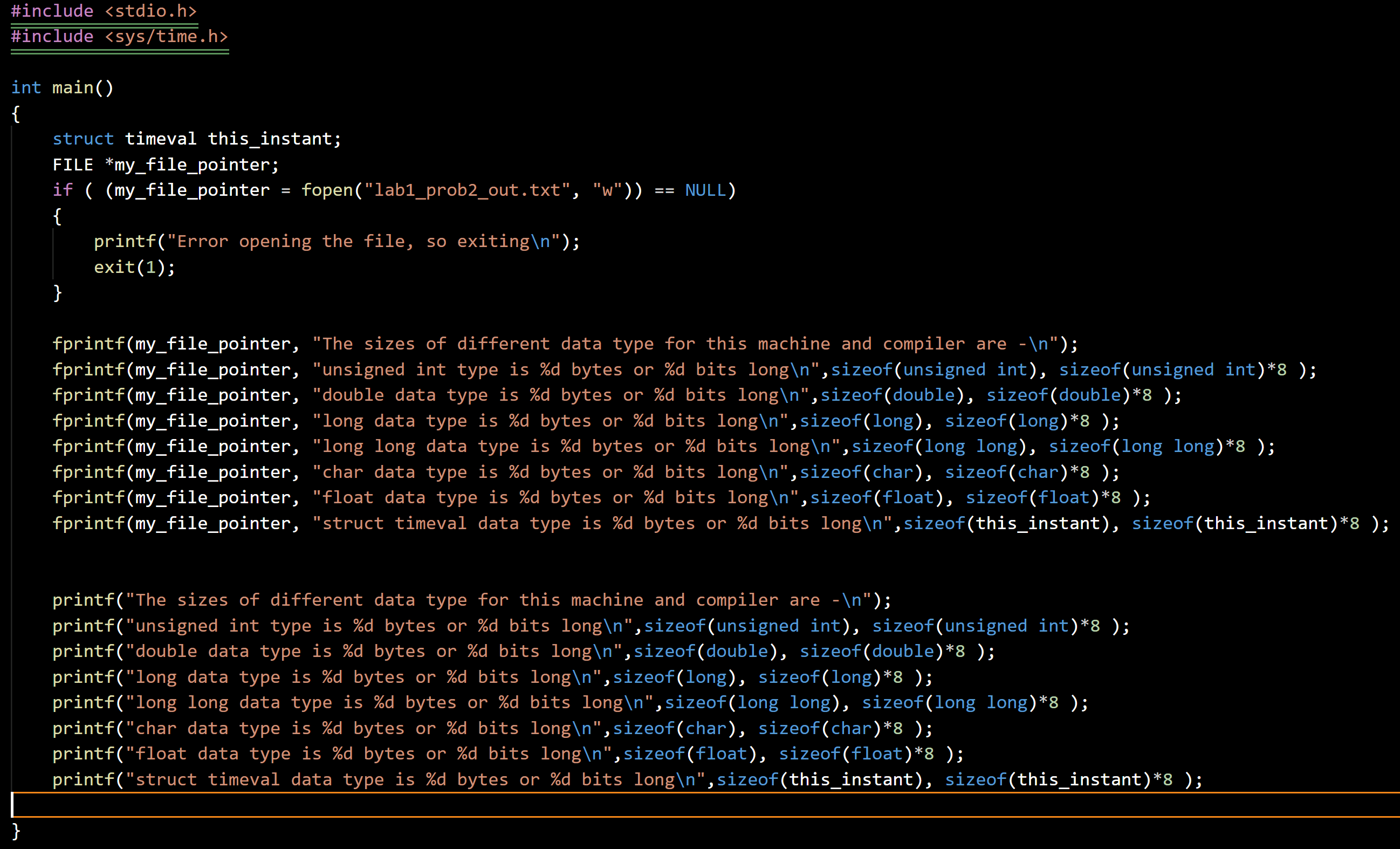
LINUX (with double)



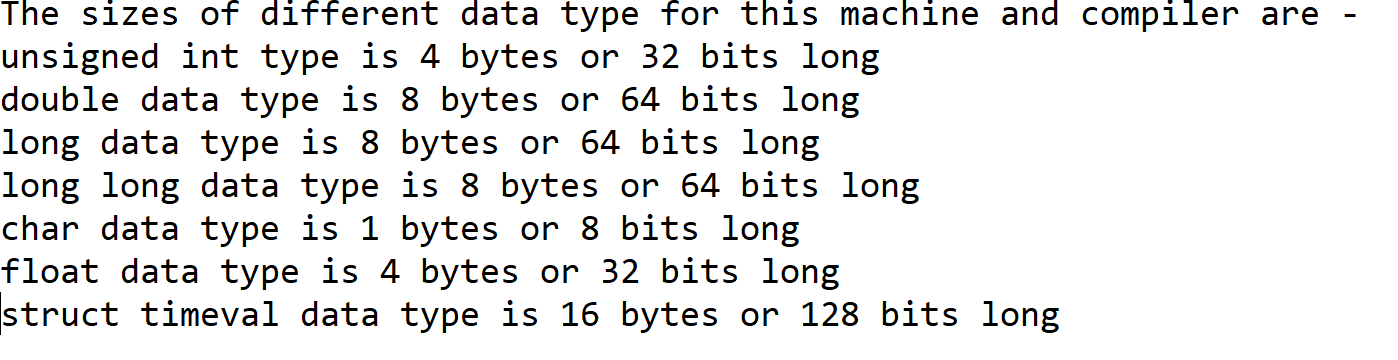
LINUX (with long int)



Exercise 3.2: Problem 2:

1. 

OUTPUT FILE:



Exercise 3.3: Problem 3:

1. 1. If (driver\_seat\_belt\_fastened == 0 && engine\_running == 1) bell ==1

2. If (door\_closed == 0 && engine\_running == 1) bell == 1

3. if (driver\_seat\_best\_fastened == 1 && door\_closed == 1 && Engine Running) bell == 0

4. if (keys\_in\_car == 1 && driver\_on\_seat == 0 && door\_closed == 1 && door\_lock\_lever == 0 or 1) door\_lock\_actuator == 0

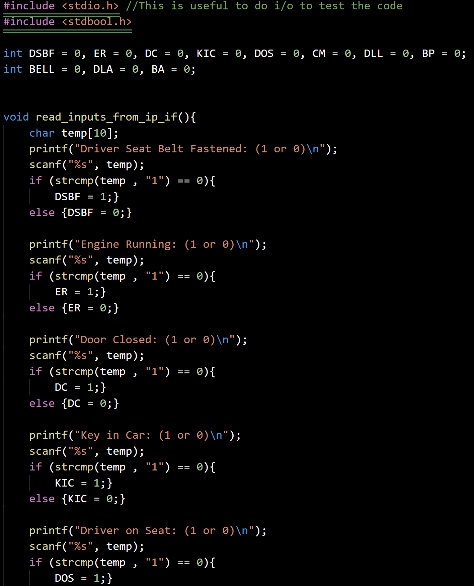
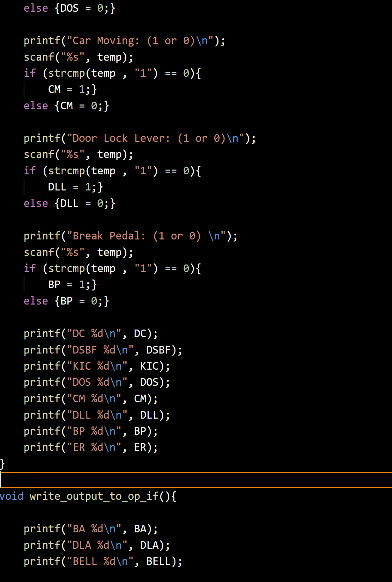
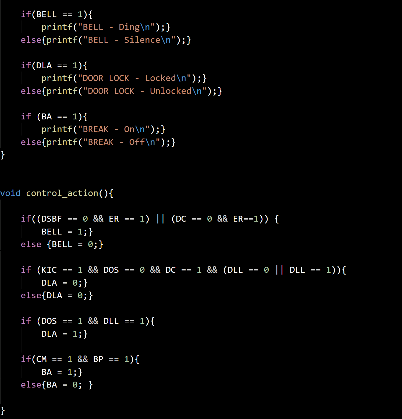
if (driver\_on\_seat == 1 && door\_lock\_lever == 1) door\_lock\_actuator ==1

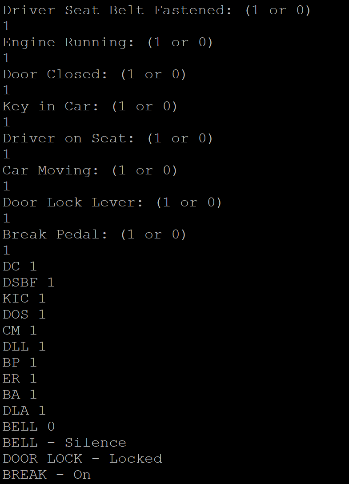
5. if (car\_moving == 1 && brake\_pedal ==1) brake actuator == 1

If (car\_moving == 0 && break\_pedal == 1) break actuator == 0

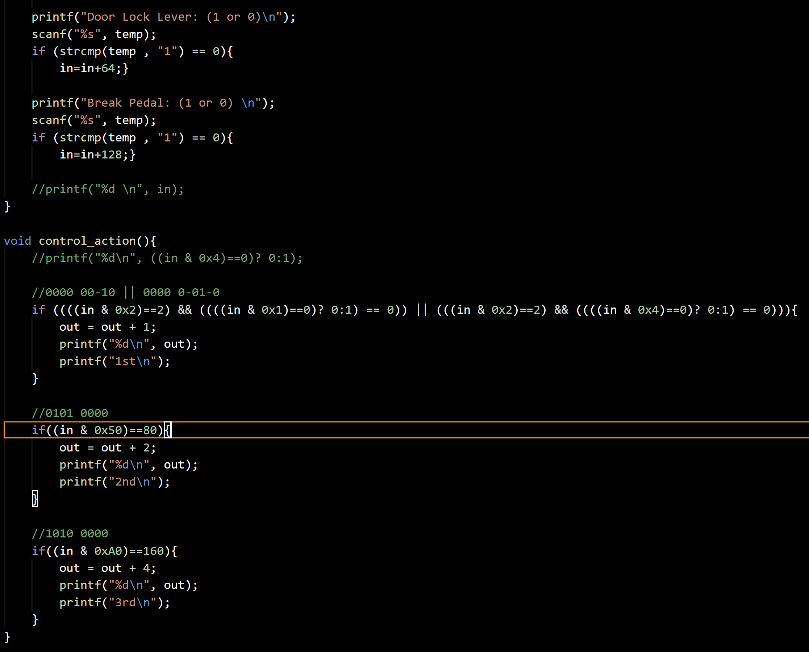
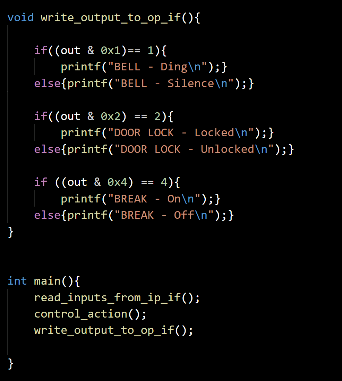
Else break\_actuator == 0

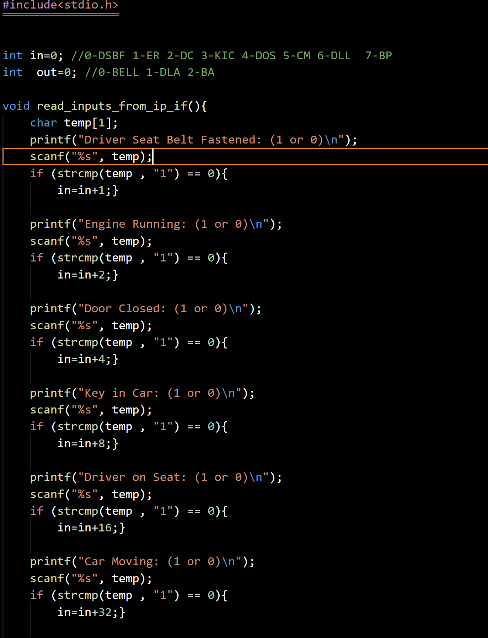
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| REQ | DOS | DSBF | ER | DC | KIC | DLC | BP | CM | BELL | DLA | BA |
| 1 | X | 0 | 1 | X | X | X | X | X | 1 | X | X |
| 2 | X | X | 1 | 0 | X | X | X | X | 1 | X | X |
| 3 | X | 1 | 1 | 1 | X | X | X | X | 0 | X | X |
| 4 | 0 | X | X | 1 | 1 | 1 | X | X | X | 0 | X |
| 4 | 0 | X | X | 1 | 1 | 0 | X | X | X | 0 | X |
| 4 | 1 | X | X | X | X | 1 | X | X | X | 1 | X |
| 5 | X | X | X | X | X | X | 1 | 1 | X | X | 1 |
| 5 | X | X | X | X | X | X | 1 | 0 | X | X | 0 |

1. 



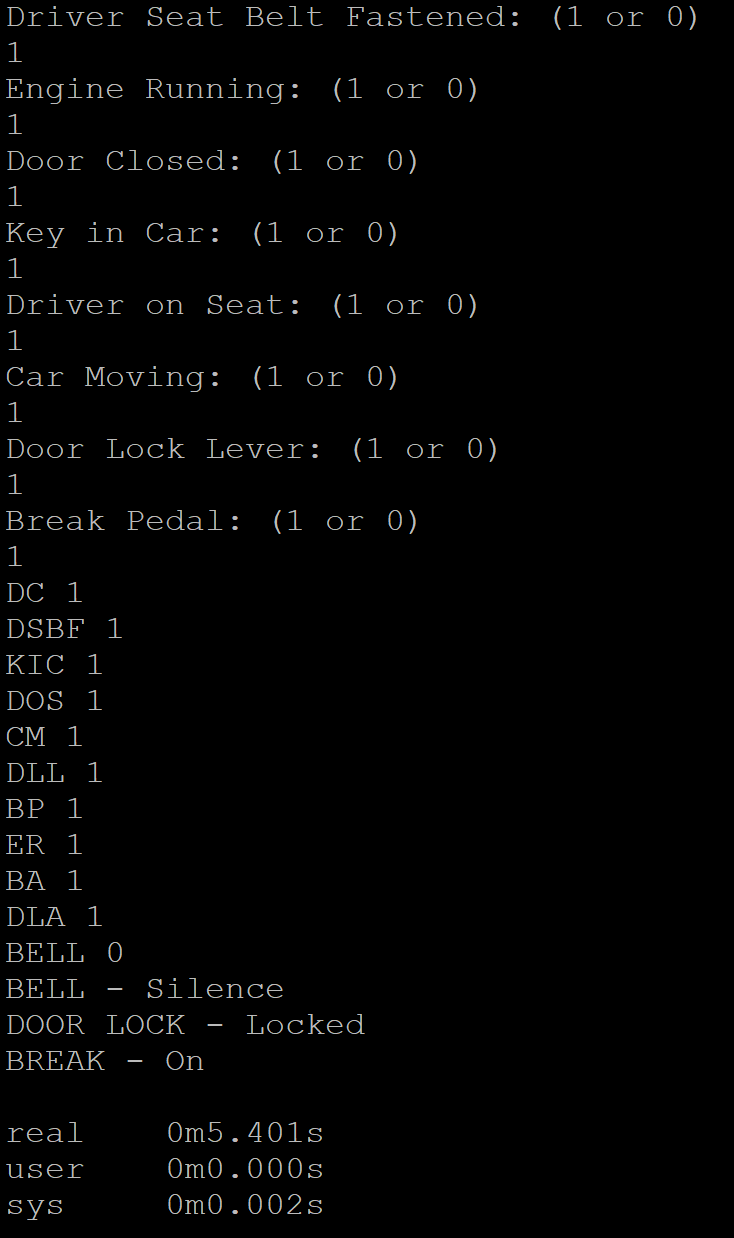
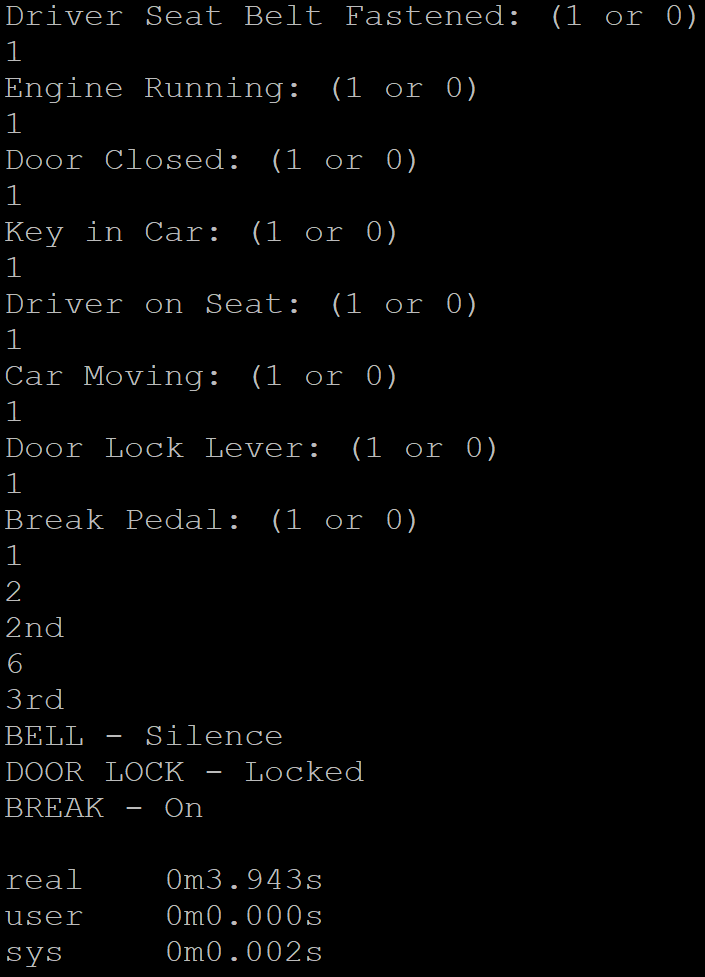
Exercise 3.4: Problem 4:

1. 



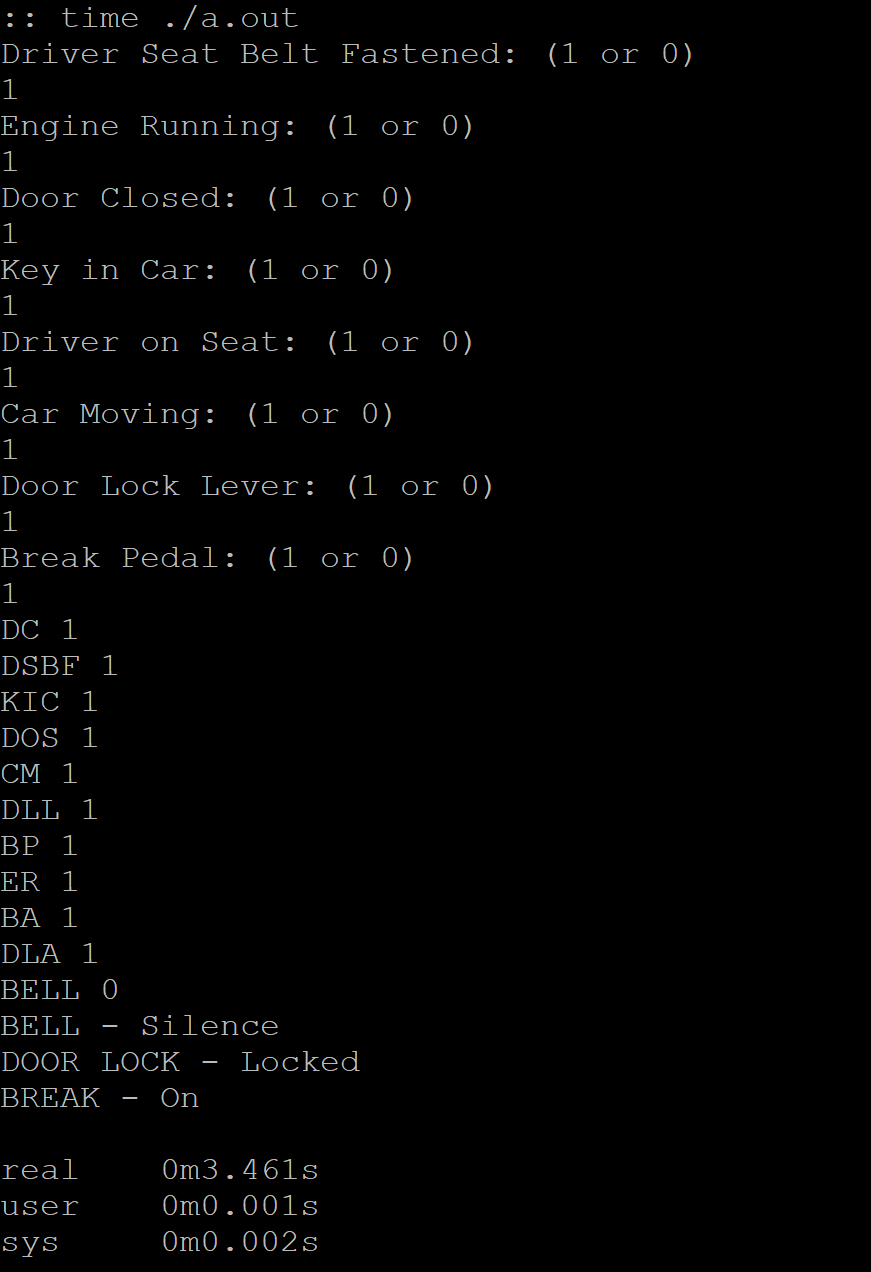
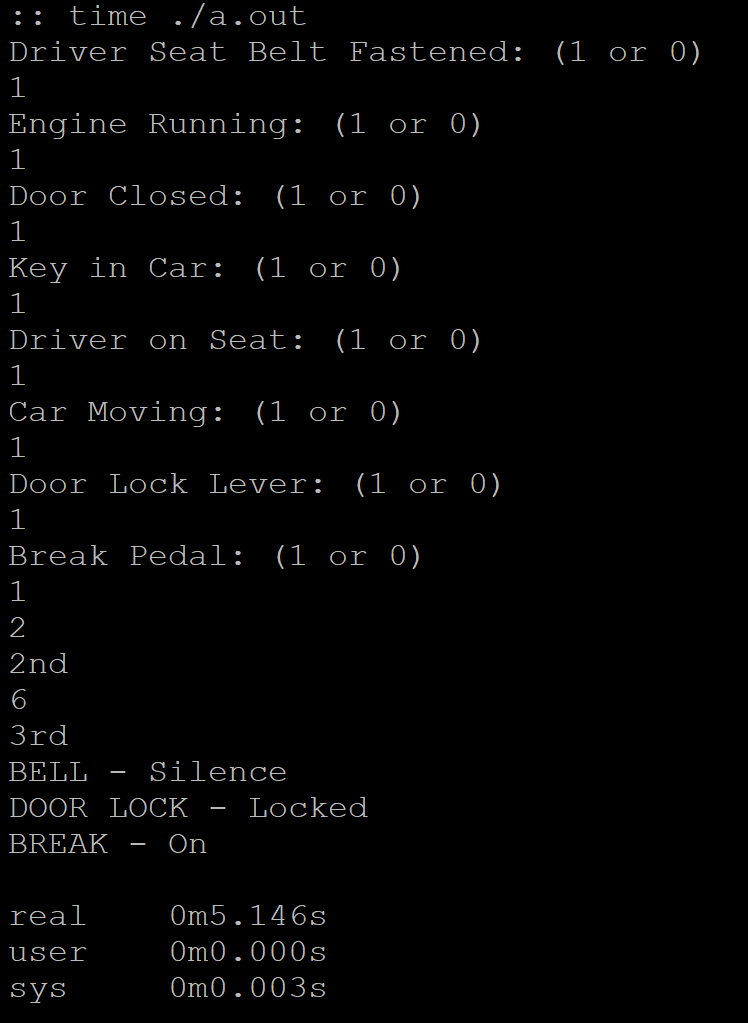
Exercise 3.5: Problem 5:

1. LINUX



PROBLEM 4 PROBLEM 3

COMPUTE



PROBLEM 4 PROBLEM 3

1. I am not convinced problem 4 is quicker than number 3. They took about the same system time with problem 4 taking a .001s longer than the one for problem 3 on the COMPUTE compiler.