



## Checkpoint 1

Checkpoint 1 checks that skill you have acquired by finishing Chapters 1 to 3. It covers basic syntax of Python as well as sequential and conditional flow of programs.

*Rules when answering coding exercises:*

- *Please answer the questions using only the basic sequential and conditional knowledge that you have learned in Chapters 1 to 3. I do not mind even you have a spaghetti logic code, just make sure you code will pass all test cases provided.*
- *Put all your programs in GitHub and I will check it from there.*
- *Put comments in your programs so other programmers will immediately know the purpose of your code.*

1. Program name: license.py

Write a python program simulating driver's license provision. Accept the age of the driver and his total number of practice hours. If driver's age is below 16, do not issue license. if 16 or above then check if number of practice hours is more than(ie. >) 200. if > 200, issue license otherwise don't.

2. Program name: jacket.py

Write a program to accept the temperature value and to tell a person to bring heavy jacket if temperature is < 20, if temperature is between 20 and 30, bring light jacket. if temperature > 30, do not bring any jacket.

3. Program name: sleep\_in.py

Write a program to accept Weekday and Vacation values. Weekday is True if it is a weekday, and the vacation is True if we are on vacation. We sleep in if it is not a weekday or we're on vacation. Return True if we sleep in.

Make sure all test cases below will pass.

`sleep_in(weekday, vacation)`

Expected	Run		
sleep_in(False, False) → True	True	OK	
sleep_in(True, False) → False	False	OK	
sleep_in(False, True) → True	True	OK	
sleep_in(True, True) → True	True	OK	

4. Program name: monkey\_trouble.py

We have two monkeys, a and b. Accept the input telling if each is smiling. We are in trouble if they are both smiling or if neither of them is smiling. Return True if we are in trouble.

Make sure all test cases below will pass.

monkey\_trouble(monkey a is smiling, monkey b is smiling)

Expected	Run		
monkey_trouble(True, True) → True	True	OK	
monkey_trouble(False, False) → True	True	OK	
monkey_trouble(True, False) → False	False	OK	
monkey_trouble(False, True) → False	False	OK	

5. Program name: sum\_double.py

Given two int values, return their sum. Unless the two values are the same, then return double their sum.

Make sure all test cases below will pass.

sum\_double(first number, second number)

Expected	Run		
sum_double(1, 2) → 3	3	OK	
sum_double(3, 2) → 5	5	OK	
sum_double(2, 2) → 8	8	OK	
sum_double(-1, 0) → -1	-1	OK	
sum_double(3, 3) → 12	12	OK	
sum_double(0, 0) → 0	0	OK	
sum_double(0, 1) → 1	1	OK	
sum_double(3, 4) → 7	7	OK	

6. Program name: parrot\_trouble.py

We have a loud talking parrot. The "hour" input is the current hour time in the range 0..23. We are in trouble if the parrot is talking(input True if talking and False if not) and the hour is before 7 or after 20. Return True if we are in trouble.

Make sure all test cases below will pass.

parrot\_trouble(talking, hour)

Expected	Run		
parrot_trouble(True, 6) → True	True	OK	
parrot_trouble(True, 7) → False	False	OK	
parrot_trouble(False, 6) → False	False	OK	
parrot_trouble(True, 21) → True	True	OK	
parrot_trouble(False, 21) → False	False	OK	
parrot_trouble(False, 20) → False	False	OK	
parrot_trouble(True, 23) → True	True	OK	
parrot_trouble(False, 23) → False	False	OK	
parrot_trouble(True, 20) → False	False	OK	
parrot_trouble(False, 12) → False	False	OK	

#### 7. Program name: cigar\_party.py

When squirrels get together for a party, they like to have cigars. A squirrel party is successful when the number of cigars is between 40 and 60, inclusive. Unless it is the weekend, in which case there is no upper bound on the number of cigars. Return True if the party with the given values is successful, or False otherwise.

Make sure all test cases below will pass.

cigar\_party(cigars, is\_weekend)

Expected	Run		
cigar_party(30, False) → False	False	OK	
cigar_party(50, False) → True	True	OK	
cigar_party(70, True) → True	True	OK	
cigar_party(30, True) → False	False	OK	
cigar_party(50, True) → True	True	OK	
cigar_party(60, False) → True	True	OK	
cigar_party(61, False) → False	False	OK	
cigar_party(40, False) → True	True	OK	
cigar_party(39, False) → False	False	OK	
cigar_party(40, True) → True	True	OK	
cigar_party(39, True) → False	False	OK	
other tests		OK	

8. Program name: caught\_speeding.py

You are driving a little too fast, and a police officer stops you. Write code to compute the result, encoded as an int value: 0=no ticket, 1=small ticket, 2=big ticket. If speed is 60 or less, the result is 0. If speed is between 61 and 80 inclusive, the result is 1. If speed is 81 or more, the result is 2. Unless it is your birthday -- on that day, your speed can be 5 higher in all cases.

Make sure all test cases below will pass.

caught\_speeding(speed, is\_birthday):

Expected	Run		
caught_speeding(60, False) → 0	0	OK	
caught_speeding(65, False) → 1	1	OK	
caught_speeding(65, True) → 0	0	OK	
caught_speeding(80, False) → 1	1	OK	
caught_speeding(85, False) → 2	2	OK	
caught_speeding(85, True) → 1	1	OK	
caught_speeding(70, False) → 1	1	OK	
caught_speeding(75, False) → 1	1	OK	
caught_speeding(75, True) → 1	1	OK	
caught_speeding(40, False) → 0	0	OK	
caught_speeding(40, True) → 0	0	OK	
caught_speeding(90, False) → 2	2	OK	
other tests		OK	

9. Program name: alarm\_clock.py

Given a day of the week encoded as 0=Sun, 1=Mon, 2=Tue, ...6=Sat, and a boolean indicating if we are on vacation, return a string of the form "7:00" indicating when the alarm clock should ring. Weekdays, the alarm should be "7:00" and on the weekend it should be "10:00". Unless we are on vacation -- then on weekdays it should be "10:00" and weekends it should be "off".

Make sure all test cases below will pass.

alarm\_clock(day, vacation)

Expected	Run		
alarm_clock(1, False) → '7:00'	'7:00'	OK	
alarm_clock(5, False) → '7:00'	'7:00'	OK	
alarm_clock(0, False) → '10:00'	'10:00'	OK	
alarm_clock(6, False) → '10:00'	'10:00'	OK	
alarm_clock(0, True) → 'off'	'off'	OK	
alarm_clock(6, True) → 'off'	'off'	OK	
alarm_clock(1, True) → '10:00'	'10:00'	OK	
alarm_clock(3, True) → '10:00'	'10:00'	OK	
alarm_clock(5, True) → '10:00'	'10:00'	OK	
other tests		OK	