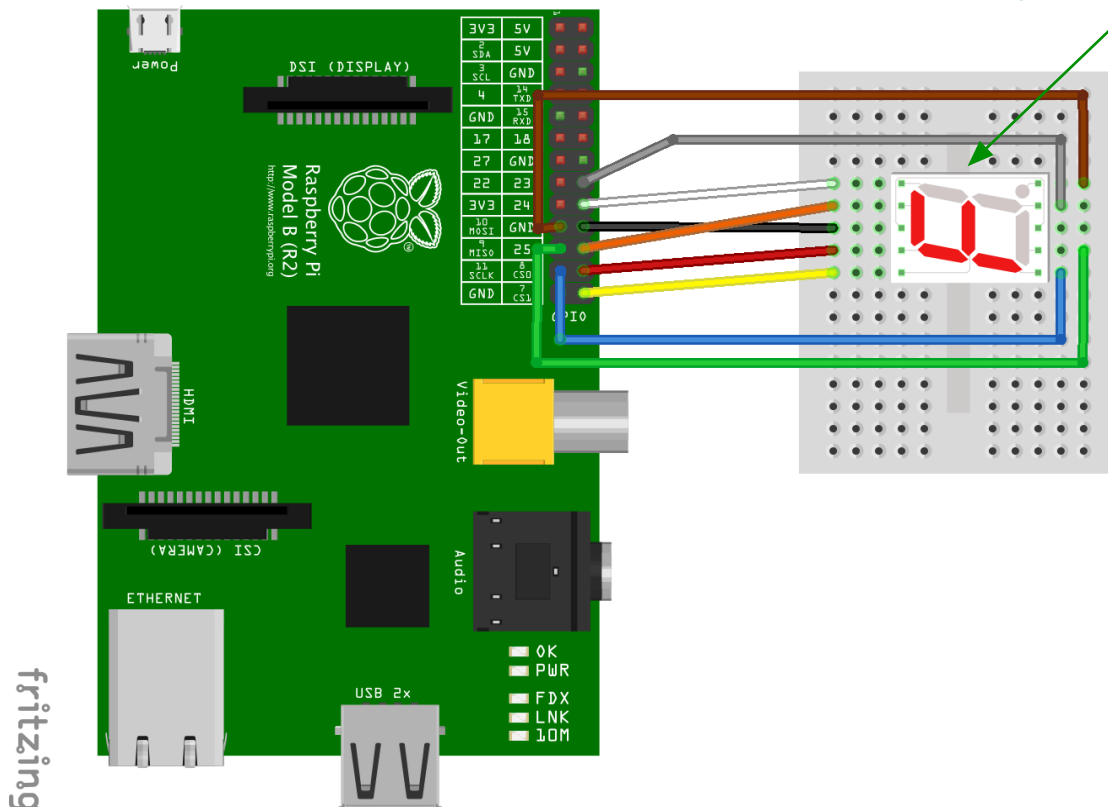
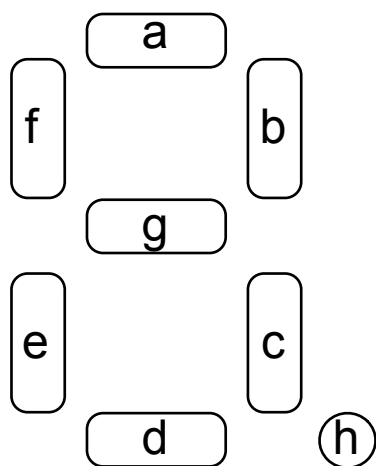


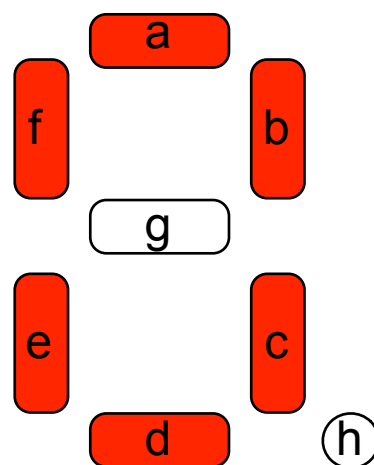
7 segment display



A seven segment display uses LEDs to show numbers. Each LED element is normally referred to by a letter, and can be switched on or off to make the correct shape.



So by turning on a, b, c, d, e, and f, we can make a zero



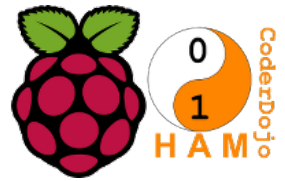
```
import RPi.GPIO as GPIO
import time
```

```
GPIO.setmode(GPIO.BOARD)
```

```
# variables to store pins for each segment
```

```
led_a = 22
led_b = 18
led_c = 16
led_d = 21
led_e = 23
led_f = 24
led_g = 26
led_h = 19
```

Use the dotted lines to help you get each block of code aligned correctly.



Can you add extra patterns for the remaining numbers (2-9) ?

```
# Design the patterns for each number
```

```
digit_zero = [led_a, led_b, led_c, led_d, led_e, led_f]
digit_one = [led_b, led_c]
```

```
# create a list of all the segment variables
```

```
leds = [led_a, led_b, led_c, led_d, led_e, led_f, led_g, led_h]
```

```
# Set all segment pins to be outputs
```

```
for pin in leds:
    GPIO.setup(pin, GPIO.OUT)
```

Remember we use [square brackets] for lists.

```
# Create simple function to turn all segments off
```

```
def all_off():
    for pin in leds:
        GPIO.output(pin, GPIO.LOW)
```

```
# Create a function to test all segments
```

```
def test_segs():
    all_off()
    for pin in leds:
        GPIO.output(pin, GPIO.HIGH)
        time.sleep(0.5)
        GPIO.output(pin, GPIO.LOW)
```

```
# create a function to display a number
```

```
def display_num(digit):
    all_off()
    for pin in digit:
        GPIO.output(pin, GPIO.HIGH)
```

```
test_segs()
time.sleep(1)
display_num(digit_zero)
time.sleep(1)
display_num(digit_one)
time.sleep(1)
```

```
GPIO.cleanup()
```

Extend and modify the code so that the leds count down from 9 to 0.
Can you use a loop and another list?