from itertools import combinations, cycle, islice, product  
  
def digit\_right\_wrong\_place(guess, digits):  
 return [  
 x  
 for x, y, z in  
 # Combine guess by given digits in the wrong place  
 zip(  
 guess,  
 # Move given digits by offset, end could be inifnite  
 islice(cycle(digits), 1, 4),  
 islice(cycle(digits), 2, 5)  
 )  
 if x == y or x == z  
 ]  
  
  
def r2(guess, digits=(2, 0, 6)):  
 matches = digit\_right\_wrong\_place(guess, digits)  
 # Convert to set to get unique matches, as it should be two digits and not the same in two places  
 return len(matches) == len(set(matches)) == 2  
  
  
rules = (  
 # One digit is right and in it's place  
 lambda guess, digits=(6, 8, 2):  
 len([True for x, y in zip(guess, digits) if x == y]) == 1,  
  
 # One digit is right but in the wrong place  
 lambda guess, digits=(6, 1, 4):  
 len(digit\_right\_wrong\_place(guess, digits)) == 1,  
  
 # Two digits is right but both are in the wrong place  
 r2,  
  
 # All digits are wrong  
 lambda guess, digits=(7, 3, 8):  
 any(x in digits for x in guess) == False,  
  
 # One digit is right but in the wrong place  
 lambda guess, digits=(3, 8, 0):  
 len(digit\_right\_wrong\_place(guess, digits)) == 1,  
  
)  
  
  
def validate(guess):  
 return all(rule(guess) for rule in rules)  
  
  
def test\_rule(rule, correct, incorrect):  
 def test\_all(guesses, answer):  
 assert all(  
 rules[rule](x) == answer  
 for x in (guesses)  
 ) == True  
  
 test\_all(correct, True)  
 test\_all(incorrect, False)  
  
  
# One digit is right and in it's place 682  
test\_rule(  
 0,  
 ((6, 0, 0), (1, 8, 0), (1, 1, 2)),  
 ((6, 8, 2), (1, 2, 0), (2, 6, 8)),  
)  
# One digit is right but in the wrong place 614  
test\_rule(  
 1,  
 ((9, 6, 9), (0, 0, 6), (1, 0, 0), (4, 0, 0)),  
 ((0, 0, 0), (6, 1, 4), (4, 6, 1), (1, 4, 6)),  
)  
# Two digits is right but both are in the wrong place 206  
test\_rule(  
 2,  
 ((0, 2, 1), (0, 6, 9), (6, 9, 2), (6, 0, 0)),  
 ((0, 0, 0), (2, 0, 6), (6, 2, 0), (0, 2, 0)),  
)  
# All digits are wrong 738  
test\_rule(  
 3,  
 ((0, 2, 1), (0, 6, 9), (6, 9, 2), (6, 0, 0)),  
 ((0, 7, 0), (3, 0, 6), (8, 2, 0), (7, 3, 8)),  
)  
# One digit is right but in the wrong place 380  
test\_rule(  
 4,  
 ((9, 3, 9), (5, 5, 3), (8, 5, 5), (0, 9, 9)),  
 ((0, 0, 0), (3, 8, 0), (0, 3, 8), (3, 3, 8)),  
)  
  
guess\_number=[guess for guess in product(range(10), repeat=3) if validate(guess)]  
print("The Three Numbers Are:")  
print("First Number:",guess\_number[0][0])  
print("Second Number:",guess\_number[0][1])  
print("Third Number:",guess\_number[0][2])