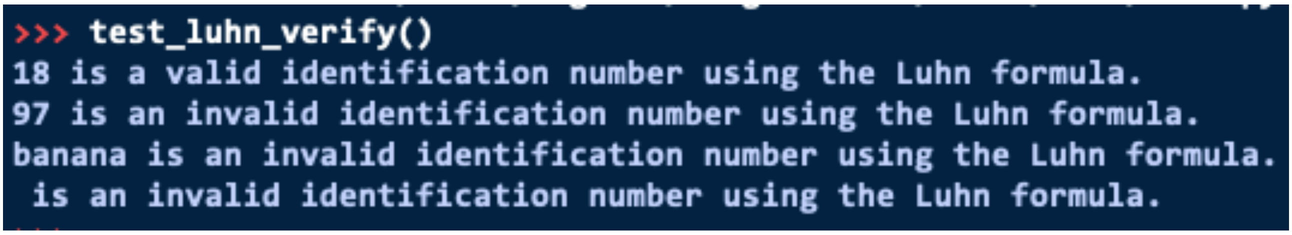
Evidence 1

```
def luhn_verify(id_number):
    if (id_number == ""):                                  #
reject empty string input: return False
        return False
    original_number = id_number[:-1][::-1]                 #
removes last digit, then flips the number
    digits_array = ['0', '1', '2', '3', '4', '5', '6', '7',
'8', '9']
    if (id_number[-1] not in digits_array):                # last
character is not a digit: return False
        return False
    check_digit = int(id_number[-1])
    sum_of_digits = 0                                      # sum
of digits in original number based on Luhn formula
    for i in range(len(original_number)):                  #
analyses every digit in the original number
        if (original_number[i] not in digits_array):       #
current character is not a digit: return False
            return False
        digit = int(original_number[i])                    #
current digit in original number
        if (i % 2 == 0):                                   # if
every other digit
            digit *= 2
            if (digit >= 10):
                digit -= 9                                 # sum
of digits of 1X = 9 - X
        sum_of_digits += digit
    if ((sum_of_digits + check_digit) % 10 == 0):          # if
valid based on Luhn formula
        return True
    return False
```

Evidence 2

```
def test_luhn_verify():
    test_cases = ['18', '97', 'banana', '']
    for num in test_cases:
        if (luhn_verify(num)):
            print("{0} is a valid identification number using
the Luhn formula.".format(num))
        else:
            print("{0} is an invalid identification number
using the Luhn formula.”.format(num))
```

Evidence 3



```
>>> test_luhn_verify()
18 is a valid identification number using the Luhn formula.
97 is an invalid identification number using the Luhn formula.
banana is an invalid identification number using the Luhn formula.
 is an invalid identification number using the Luhn formula.
```

id_number	Purpose of test	Expected output
“18”	Test whether code works when number is valid	18 is a valid identification number using the Luhn formula.
“97”	Test whether code works when number is invalid	97 is an invalid identification number using the Luhn formula.
“banana”	Validate parameters with non-digits	banana is an invalid identification number using the Luhn formula.
“”	Test for no input	is an invalid identification number using the Luhn formula.

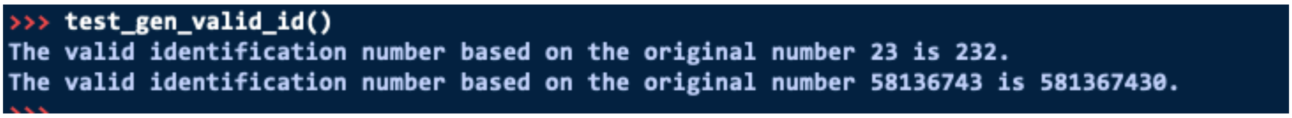
Evidence 4

```
def gen_valid_id(number):
    flipped_number = number[::-1]                       #
flips the number to make it easier to analyse
    sum_of_digits = 0                                   # sum
of digits in original number based on Luhn formula
    for i in range(len(flipped_number)):                #
analyses every digit in the original number
        digit = int(flipped_number[i])                  #
current digit in original number
        if (i % 2 == 0):                                # if
every other digit
            digit *= 2
            if (digit >= 10):
                digit -= 9                              # sum
of digits of 1X = 9 - X
        sum_of_digits += digit
    check_digit = (10 - (sum_of_digits % 10)) % 10      #
finds appropriate check digit
    return number + str(check_digit)
```

Evidence 5

```
def test_gen_valid_id():
    test_cases = ['23', '58136743']
    for num in test_cases:
        print("The valid identification number based on the
original number {0} is {1}".format(num, gen_valid_id(num)))
```

Evidence 6



```
>>> test_gen_valid_id()
The valid identification number based on the original number 23 is 232.
The valid identification number based on the original number 58136743 is 581367430.
>>>
```