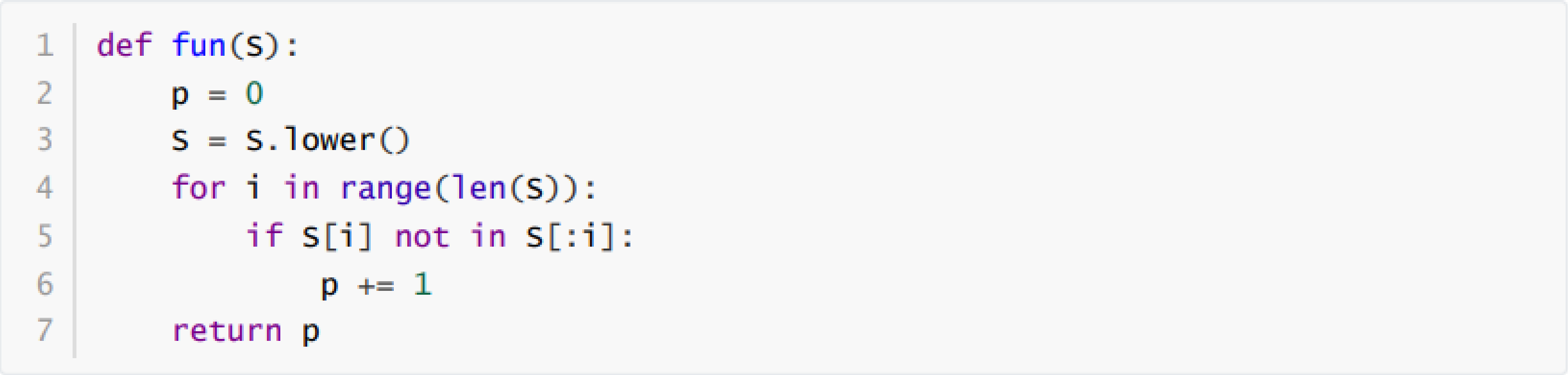
Note: This assignment will be evaluated after the deadline passes. You will get your score 48 hrs after the deadline. Until then the score will be shown as Zero.

1) *1 point*



```
def fun(S):
    p = 0
    S = S.lower()
    for i in range(len(S)):
        if S[i] not in S[:i]:
            p += 1
    return p
```

S is a non-empty string of English letters without any space. What fun(S) will return after execution of the above code?

○ Total number of letters in the string S .

● Total number of distinct letters in the string S .

○ Total number of letters that are repeated in the string S more than one time.

○ Difference of total letters in the string S and distinct letters in the string S .

2) Which of the following is/are valid reason(s) for NameError exception? [MSQ] *1 point*

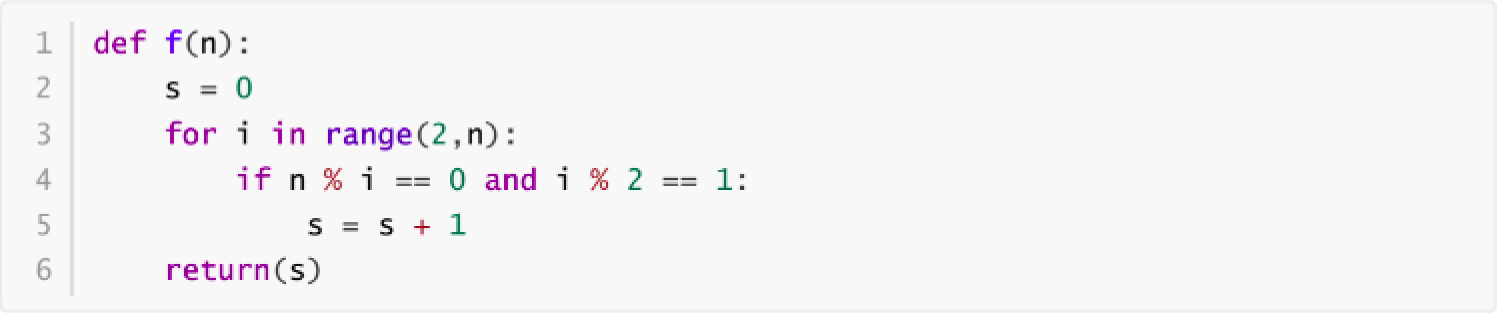
☑ Variable is not defined.

☑ Calling a function before declaration.

☑ Misspelled built-in functions name

□ Variables are defined globally in the program .

3)



```python
def f(n):
    s = 0
    for i in range(2,n):
        if n % i == 0 and i % 2 == 1:
            s = s + 1
    return(s)
```

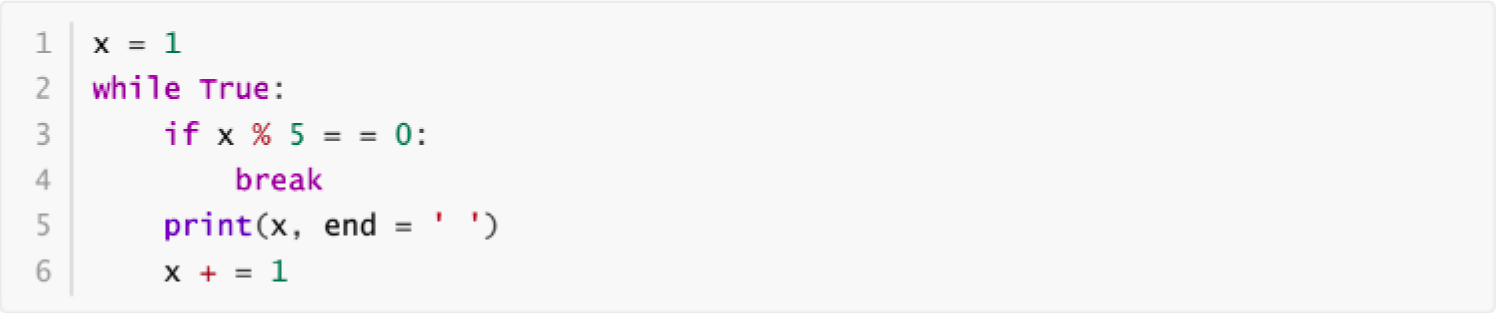
What is f(60) - f(59) , given the definition of f above?

3

1 point

4)

1 point



```python
x = 1
while True:
    if x % 5 = = 0:
        break
    print(x, end = ' ')
    x + = 1
```

What will be the output of the above code-snippet?

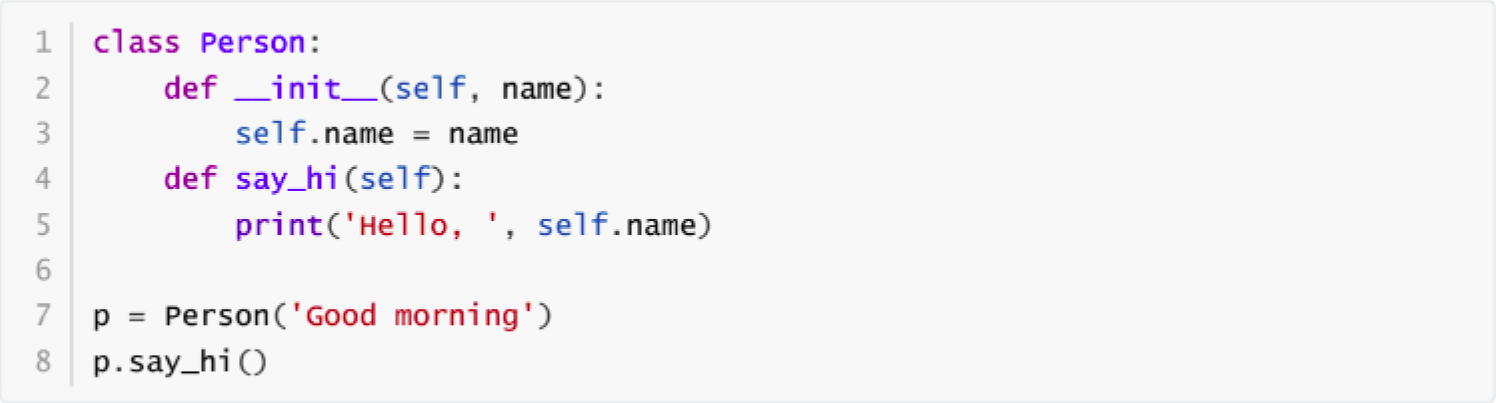
● Syntax error

○ 2 1

○ 0 3 1

○ None of these

5) ***1 point***



```
class Person:
    def __init__(self, name):
        self.name = name
    def say_hi(self):
        print('Hello, ', self.name)

p = Person('Good morning')
p.say_hi()
```

What will be the output of the above code-snippet?

○ Good morning

● Hello, Good morning

○ Hello Good morning

○ Good

6)

```python
a = [1, 2, 3]
try:
    print ("Second element = %d" %(a[1]))
    print ("Fourth element = %d" %(a[3]))
except:
    print ("An error occurred")
```

1 point

What will be the output of the above code-snippet?

○

```
Second element = 2
```

○

```
An error occured
```

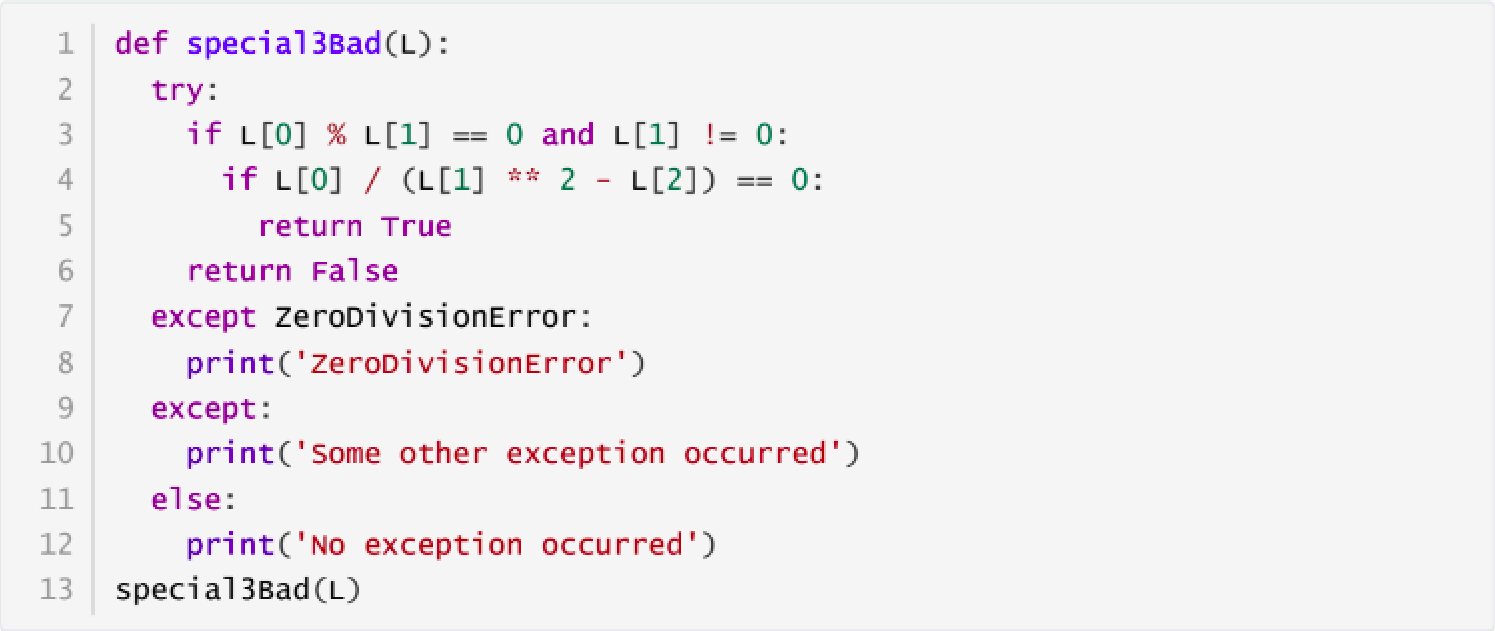
●

```
Second element = 2
An error occurred
```

○

```
Second element = 2
Fourth element = 3
An error occurred
```

7) ***1 point***

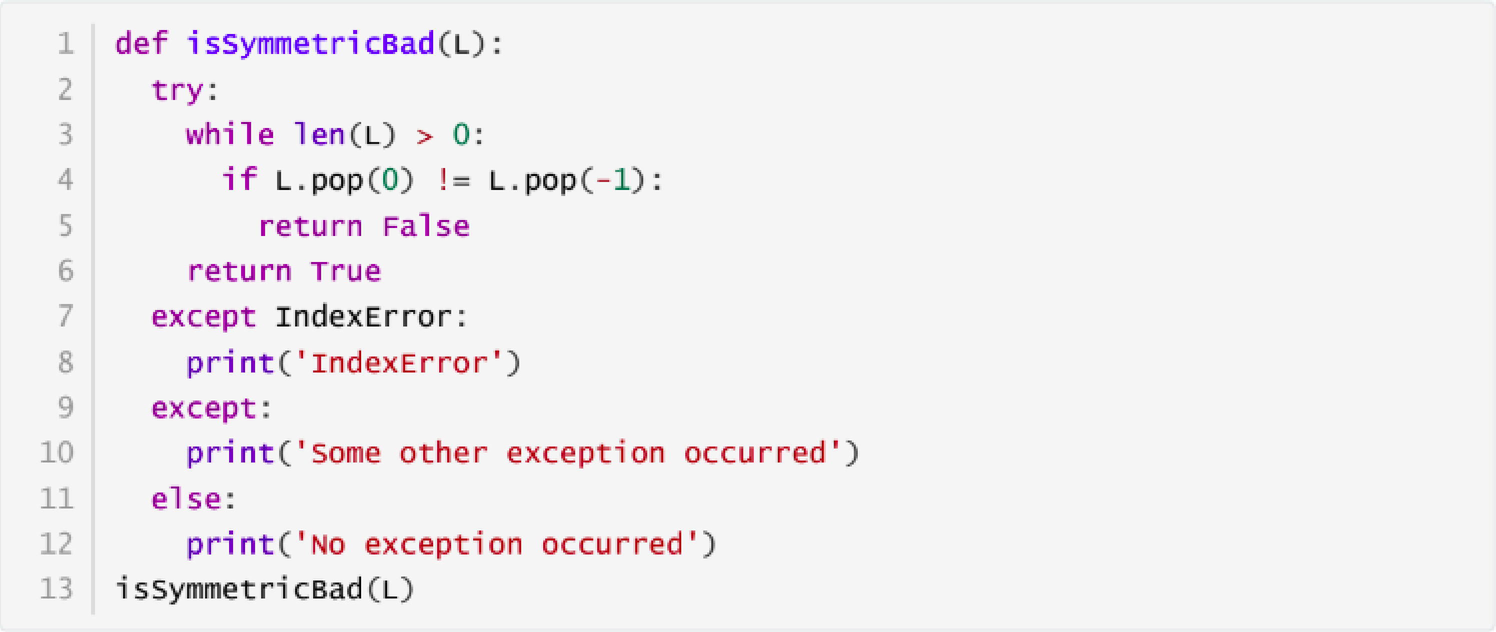


```
def special3Bad(L):
  try:
    if L[0] % L[1] == 0 and L[1] != 0:
      if L[0] / (L[1] ** 2 - L[2]) == 0:
        return True
    return False
  except ZeroDivisionError:
    print('ZeroDivisionError')
  except:
    print('Some other exception occurred')
  else:
    print('No exception occurred')
special3Bad(L)
```

Given above is a function that checks whether a list satisfies some property. There is an error in this function. Select the list(s) L = [n1, n2, n3] , where n1 , n2 and n3 are all integers, for which special3Bad(L) produces a ZeroDivisionError exception. [MSQ]

- □ L = [4, 2, 8]
- ☑ L = [4, 2, 4]
- ☑ L = [8, 4, 16]
- ☑ L = [48, 6, 36]
- □ L = [44, 6, 36]

8) *1 point*



```python
def isSymmetricBad(L):
  try:
    while len(L) > 0:
      if L.pop(0) != L.pop(-1):
        return False
    return True
  except IndexError:
    print('IndexError')
  except:
    print('Some other exception occurred')
  else:
    print('No exception occurred')
isSymmetricBad(L)
```

Given above is a function to check whether a list is a palindrome. There is an error in this function. Select the list(s) L = [n1, n2,..., n2, n1] , for which isSymmetricBad(L) produces an IndexError exception. [MSQ]

☑ L = [1, 2, 3, 4, 3, 2, 1]

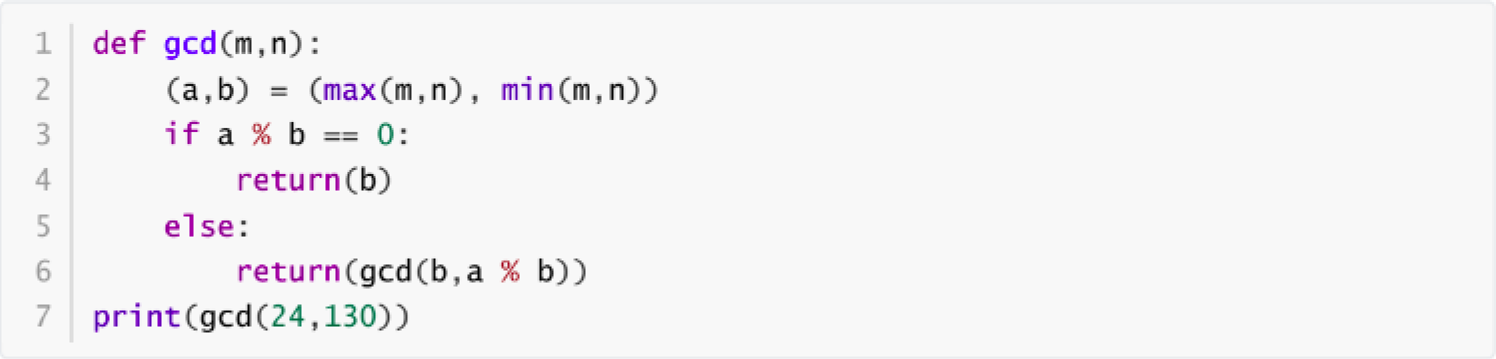
☐ L = [2, 2, 2, 2, 2, 2]

☑ L = [1, 1, 1, 1, 1, 1, 1]

☑ L = [8]

☐ L = [2, 4, 6]

9)



```python
def gcd(m,n):
    (a,b) = (max(m,n), min(m,n))
    if a % b == 0:
        return(b)
    else:
        return(gcd(b,a % b))
print(gcd(24,130))
```

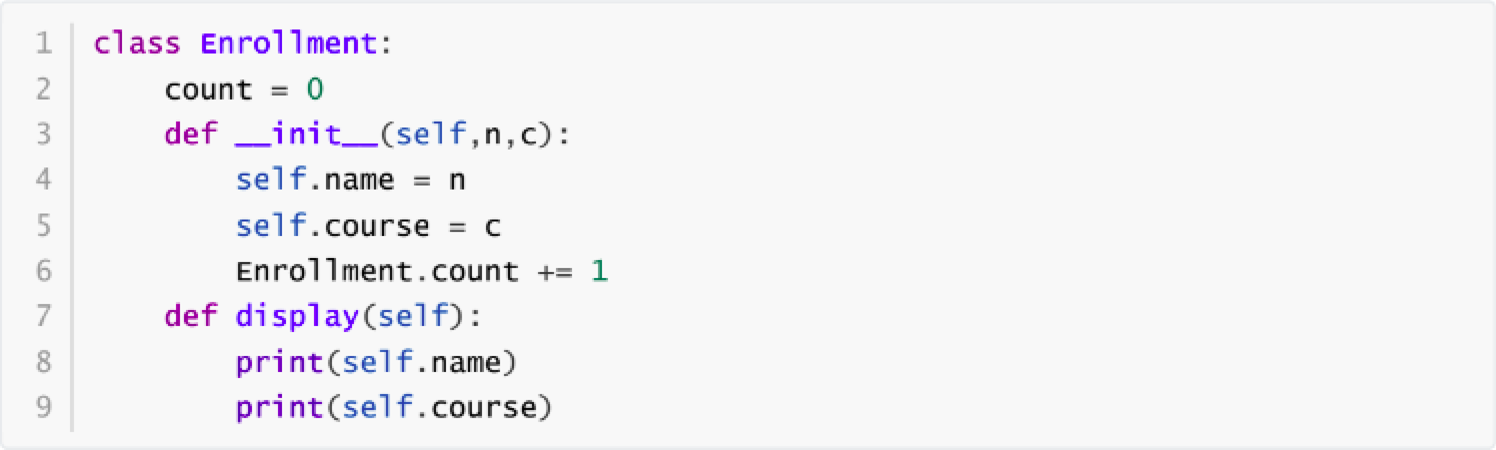
How many times gcd() function will be called?

Note: Ignore the first call given in the code.

3

1 point

10) ***1 point***



```python
class Enrollment:
    count = 0
    def __init__(self,n,c):
        self.name = n
        self.course = c
        Enrollment.count += 1
    def display(self):
        print(self.name)
        print(self.course)
```

Which of the following option(s) is/are correct about the given code? [MSQ]

☐ count , name and course are object variables.

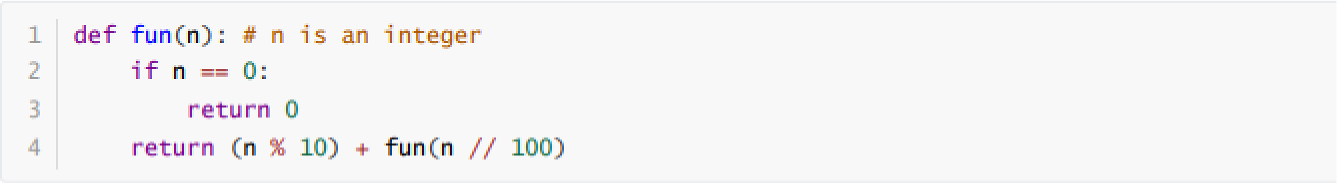
□ name and course are class variable and count is an object variable.

☑ name and course are object variables, and count is a class variable.

□ count , name and course are class variables.

☑ count represents the number of objects created for class Enrollment

11) *1 point*



```
def fun(n): # n is an integer
    if n == 0:
        return 0
    return (n % 10) + fun(n // 100)
```

What does the above code compute?

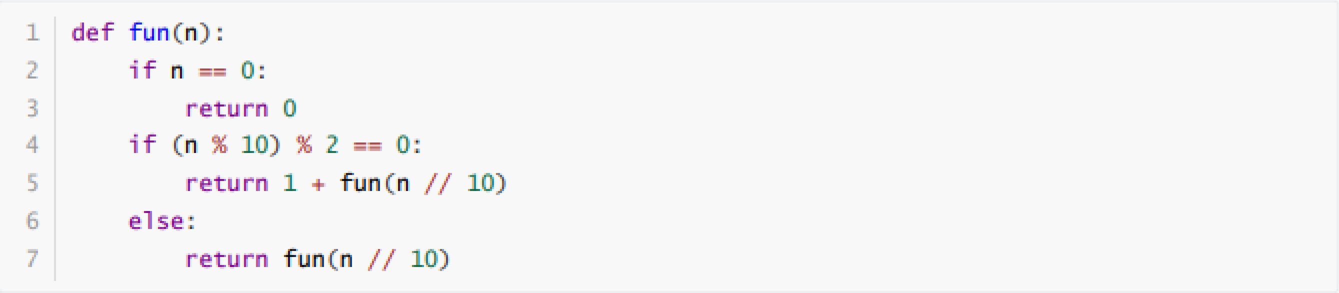
○ The sum of all digits

● The sum of alternate digits starting from the last digit

○ The sum of digits at even indexed positions (from left, 0-based)

○ The sum of alternate digits starting from the first digit from left

12)



```
def fun(n):
    if n == 0:
        return 0
    if (n % 10) % 2 == 0:
        return 1 + fun(n // 10)
    else:
        return fun(n // 10)
```

1 point

What is the final value returned by the function?

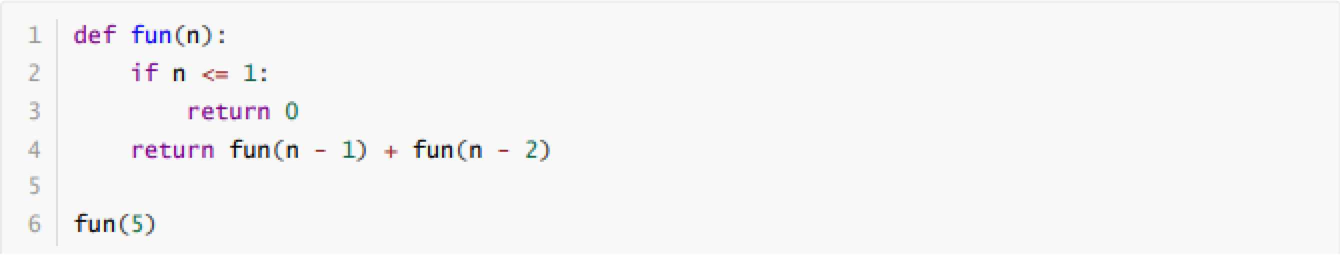
○ Number of even digits

● Sum of even digits

○ Sum of odd digits

○ Number of odd digits

13) Consider the code given below



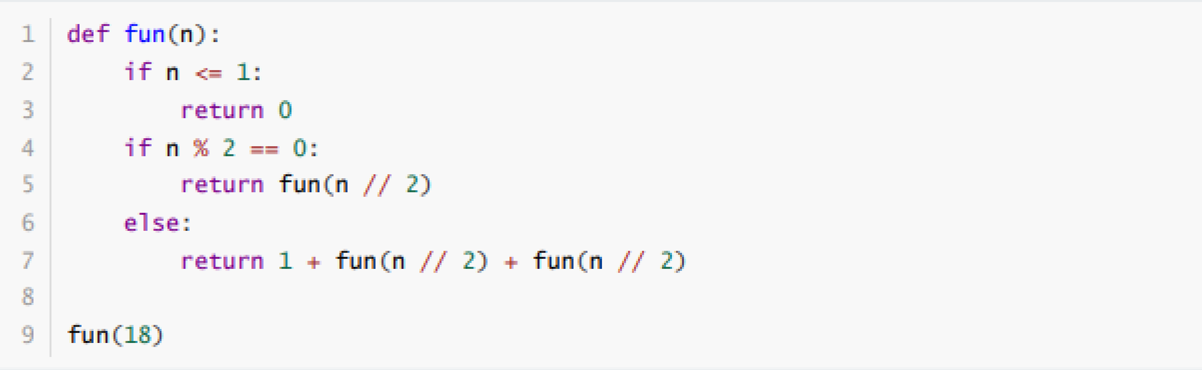
```
def fun(n):
    if n <= 1:
        return 0
    return fun(n - 1) + fun(n - 2)

fun(5)
```

How many recursive calls are made (excluding the initial call to fun(5)) ?

14

1 point

14) Consider the code given below.



```python
def fun(n):
    if n <= 1:
        return 0
    if n % 2 == 0:
        return fun(n // 2)
    else:
        return 1 + fun(n // 2) + fun(n // 2)

fun(18)
```

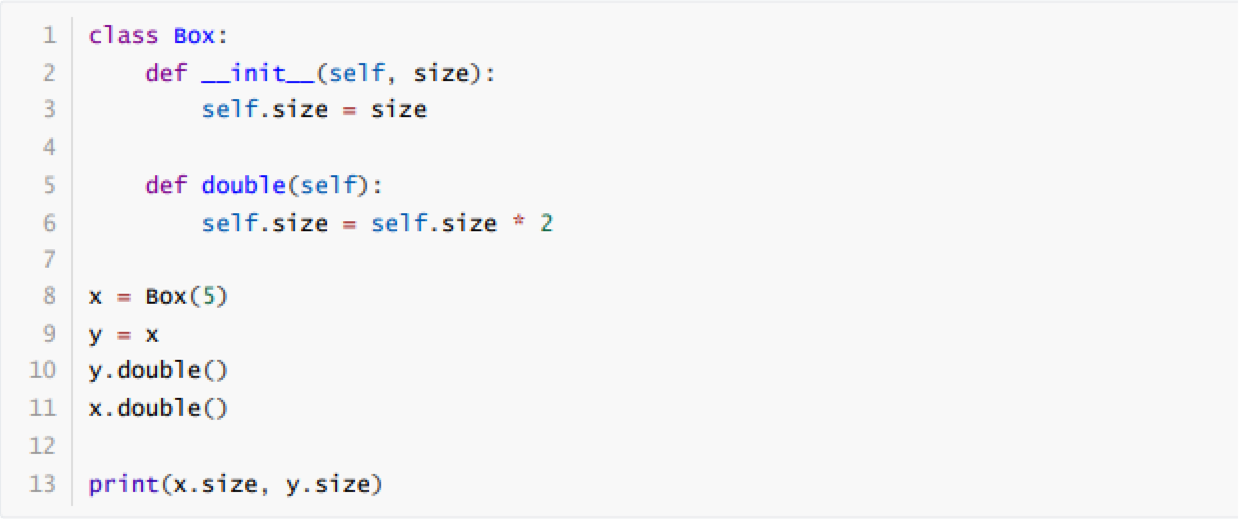
How many recursive calls (excluding the initial fun(18)) are made?

7

1 point

15) Consider the code given below.

1 point



```python
class Box:
    def __init__(self, size):
        self.size = size

    def double(self):
        self.size = self.size * 2

x = Box(5)
y = x
y.double()
x.double()

print(x.size, y.size)
```

Select all correct statements:

☑ Output will be 20 20

□ Output will be 10 10

☑ x and y refer to the same object

□ x and y are independent copies of objects

☑ Method double() modifies the object in-place