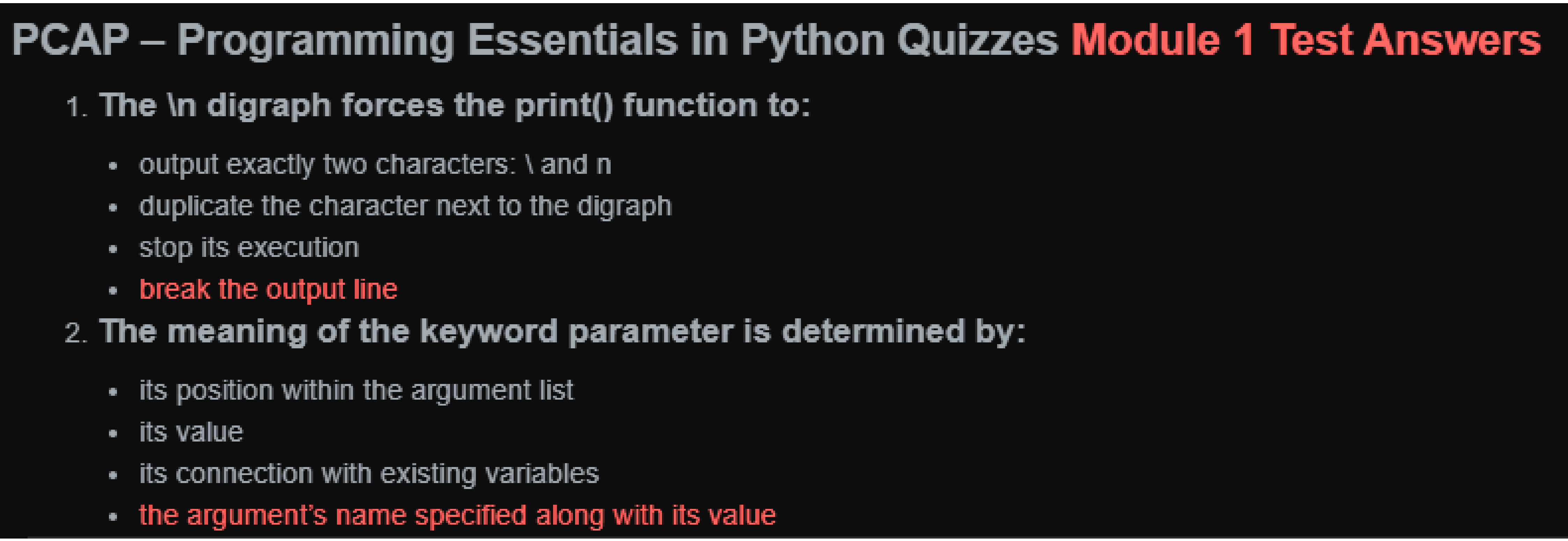
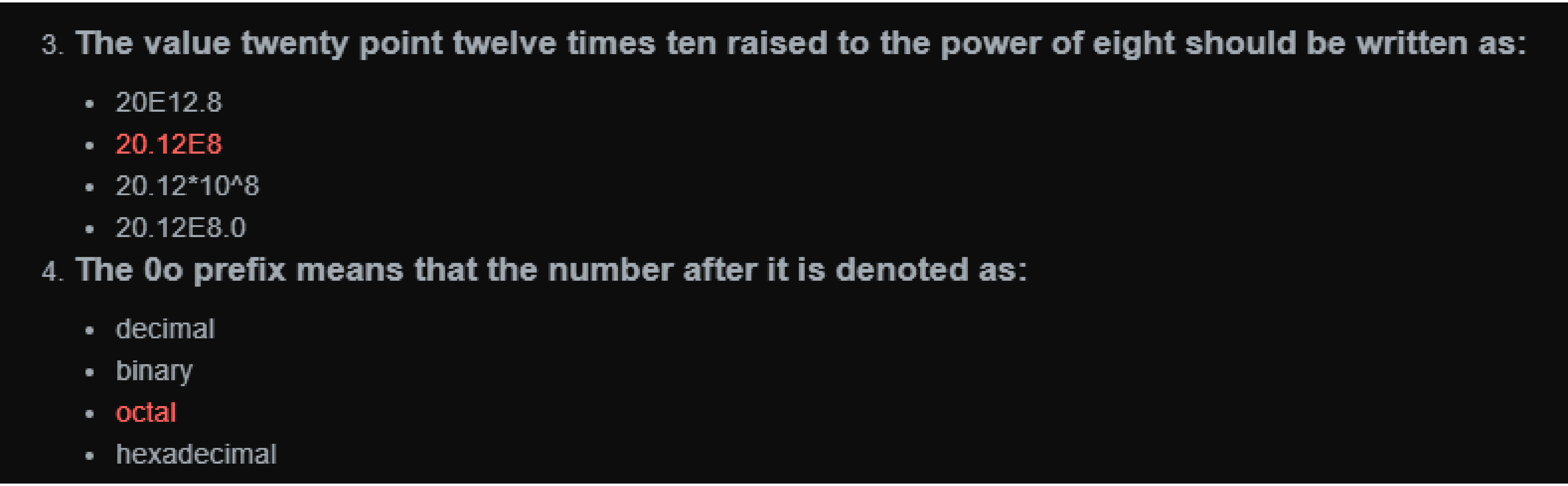
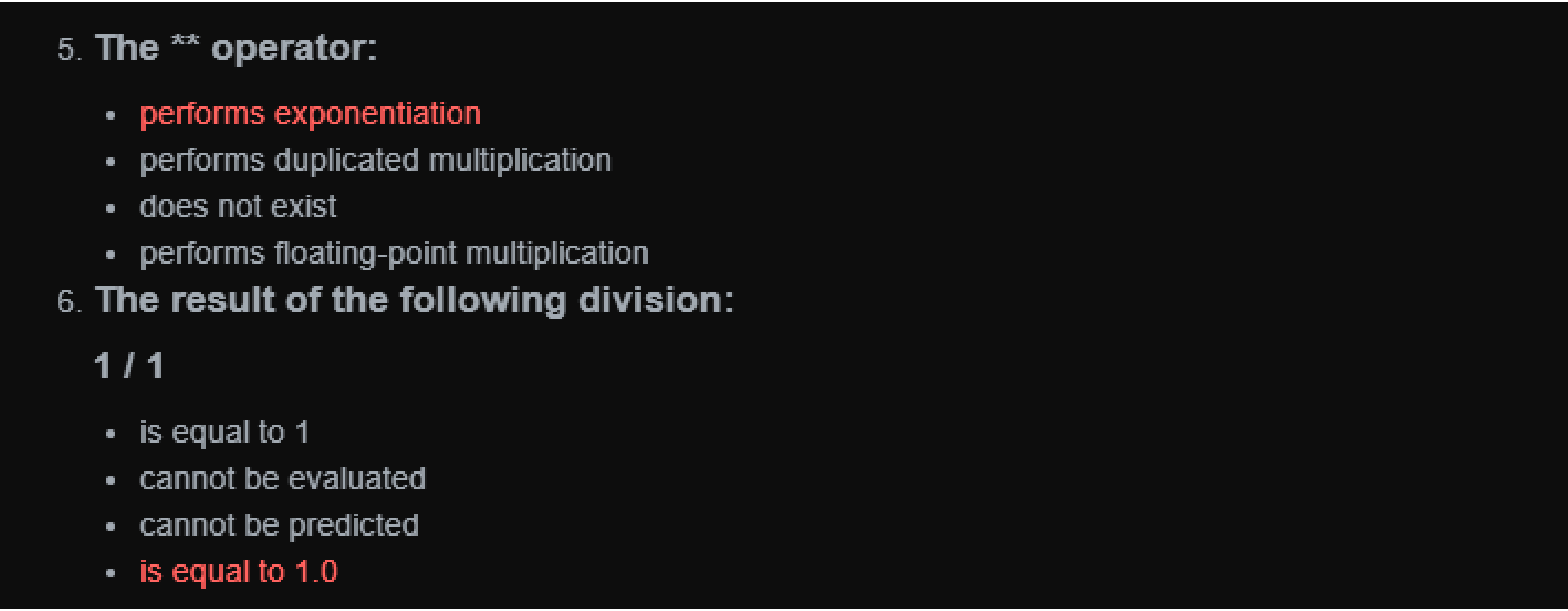


PCAP – Programming Essentials in Python Quizzes Module 1 Test Answers

1. **The \n digraph forces the print() function to:**
   - output exactly two characters: \ and n
   - duplicate the character next to the digraph
   - stop its execution
   - [✓ break the output line]
2. **The meaning of the keyword parameter is determined by:**
   - its position within the argument list
   - its value
   - its connection with existing variables
   - [✓ the argument’s name specified along with its value]



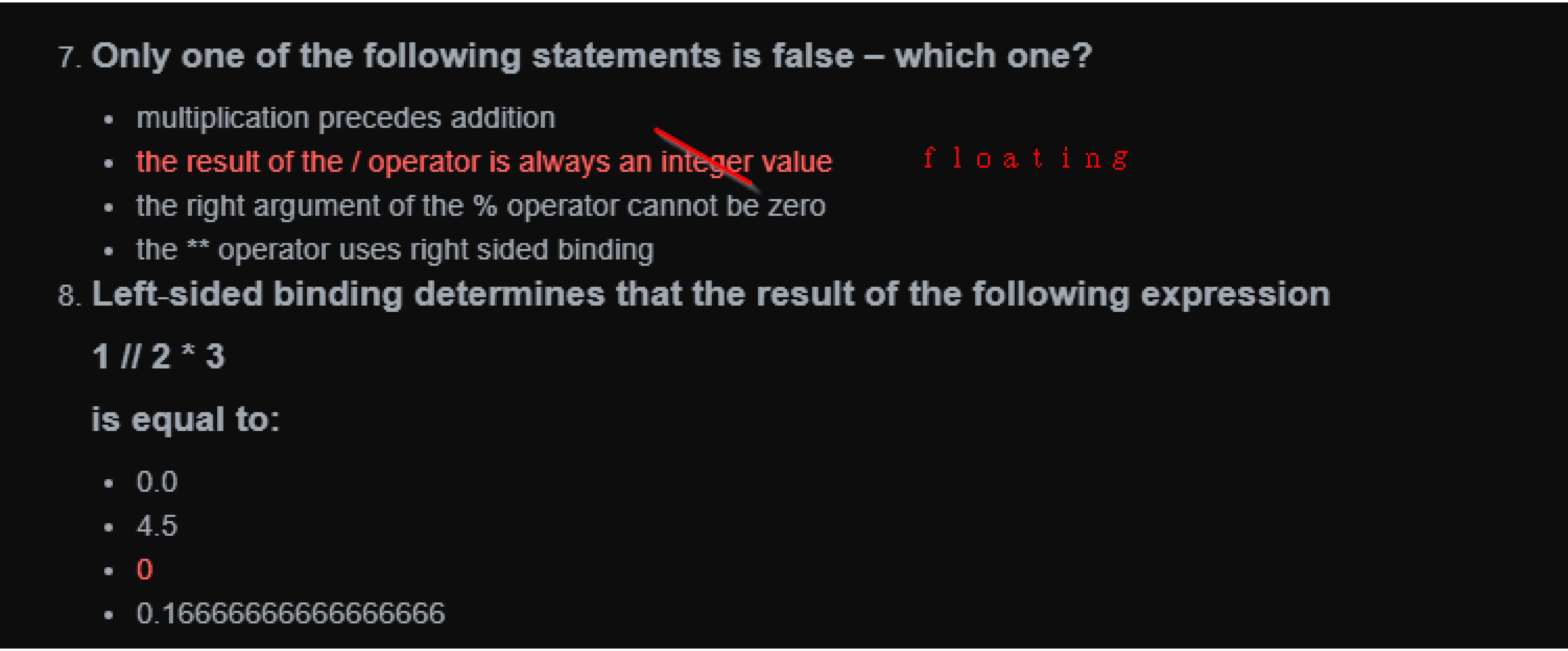
3. **The value twenty point twelve times ten raised to the power of eight should be written as:**
    - 20E12.8
    - [✓ 20.12E8]
    - 20.12*10^8
    - 20.12E8.0
4. **The 0o prefix means that the number after it is denoted as:**
    - decimal
    - binary
    - [✓ octal]
    - hexadecimal



5. **The ** operator:**
   - [✓ performs exponentiation]
   - performs duplicated multiplication
   - does not exist
   - performs floating-point multiplication
6. **The result of the following division:**

   1 / 1

   - is equal to 1
   - cannot be evaluated
   - cannot be predicted
   - [✓ is equal to 1.0]



7. **Only one of the following statements is false – which one?**

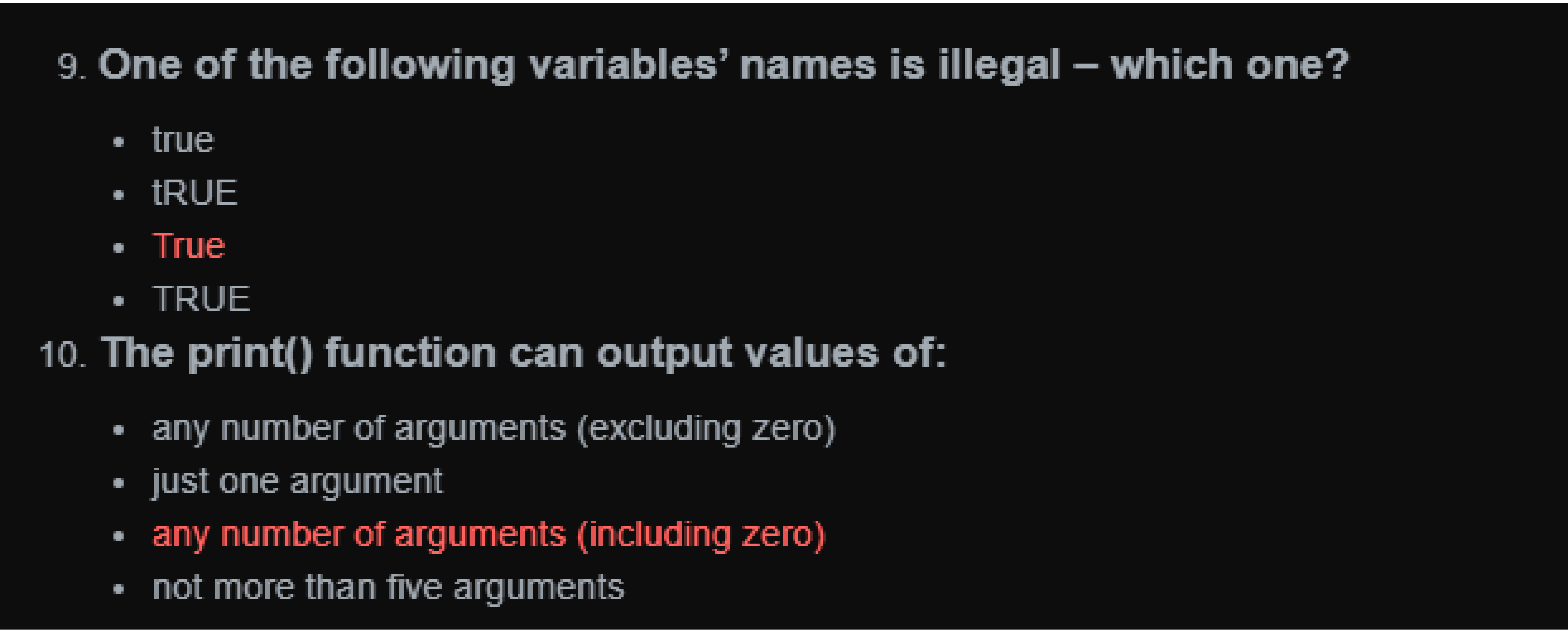
- multiplication precedes addition
- [✓ the result of the / operator is always an ~~integer~~ value] f l o a t i n g
- the right argument of the % operator cannot be zero
- the ** operator uses right sided binding

8. **Left-sided binding determines that the result of the following expression**

   1 // 2 * 3

   is equal to:

- 0.0
- 4.5
- [✓ 0]
- 0.16666666666666666

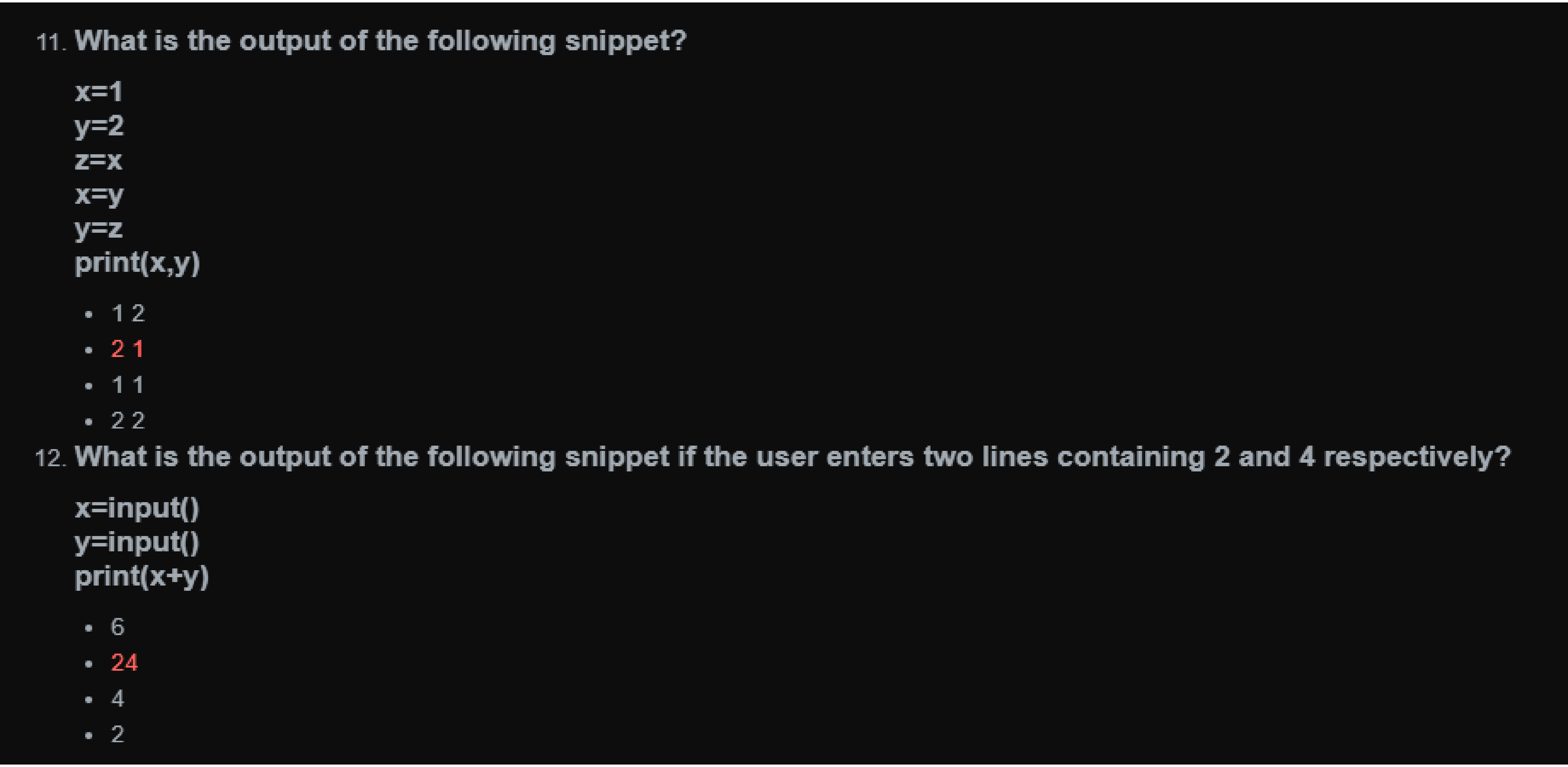


9. **One of the following variables' names is illegal – which one?**

- true
- tRUE
- [✓ True]
- TRUE

10. **The print() function can output values of:**

- any number of arguments (excluding zero)
- just one argument
- [✓ any number of arguments (including zero)]
- not more than five arguments



11. **What is the output of the following snippet?**

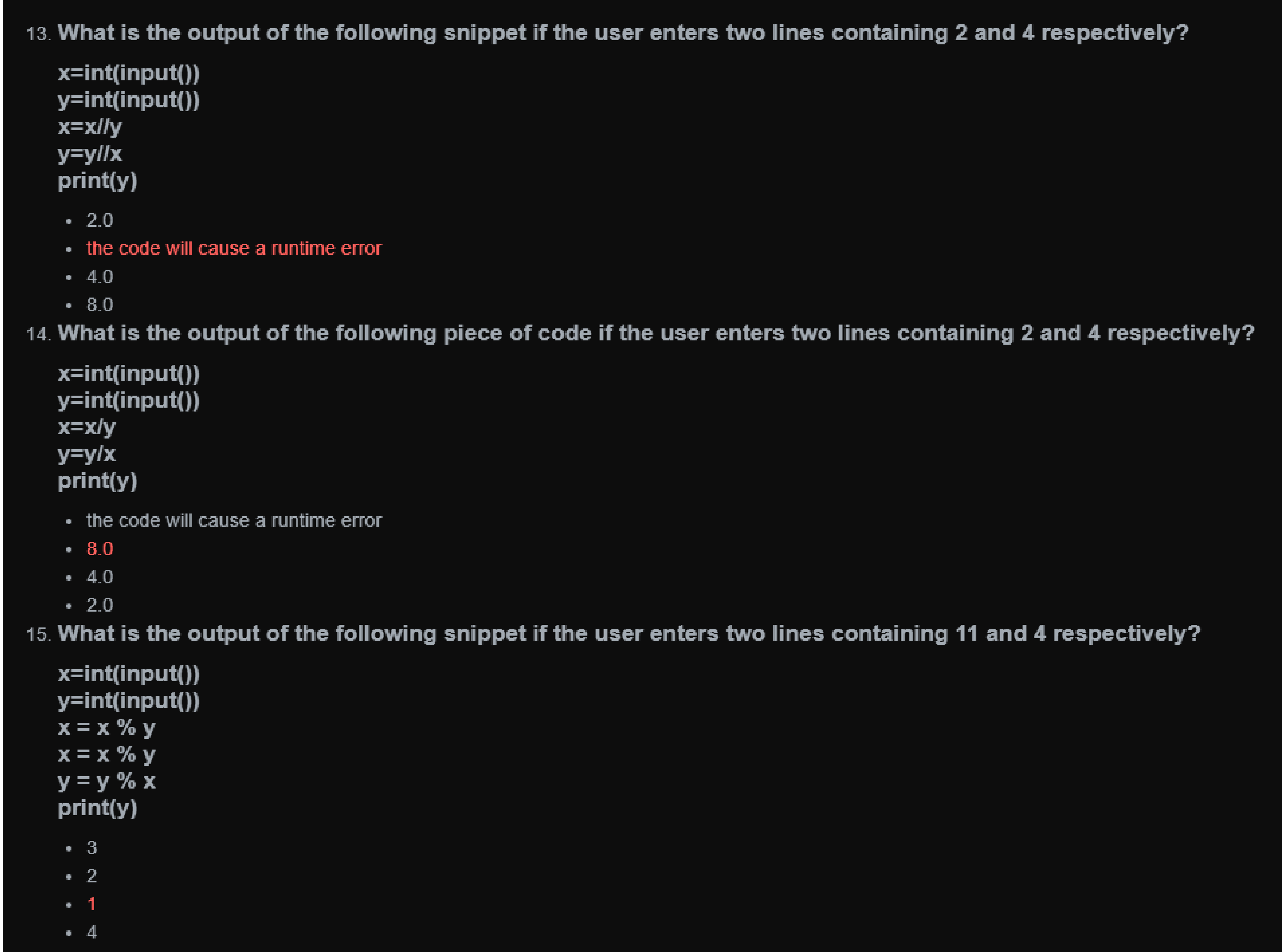
```
x=1
y=2
z=x
x=y
y=z
print(x,y)
```

- 1 2
- [✓ 2 1]
- 1 1
- 2 2

12. **What is the output of the following snippet if the user enters two lines containing 2 and 4 respectively?**

```
x=input()
y=input()
print(x+y)
```

- 6
- [✓ 24]
- 4
- 2



13. **What is the output of the following snippet if the user enters two lines containing 2 and 4 respectively?**

```
x=int(input())
y=int(input())
x=x//y
y=y//x
print(y)
```

- 2.0
- [✓ the code will cause a runtime error]
- 4.0
- 8.0

14. **What is the output of the following piece of code if the user enters two lines containing 2 and 4 respectively?**

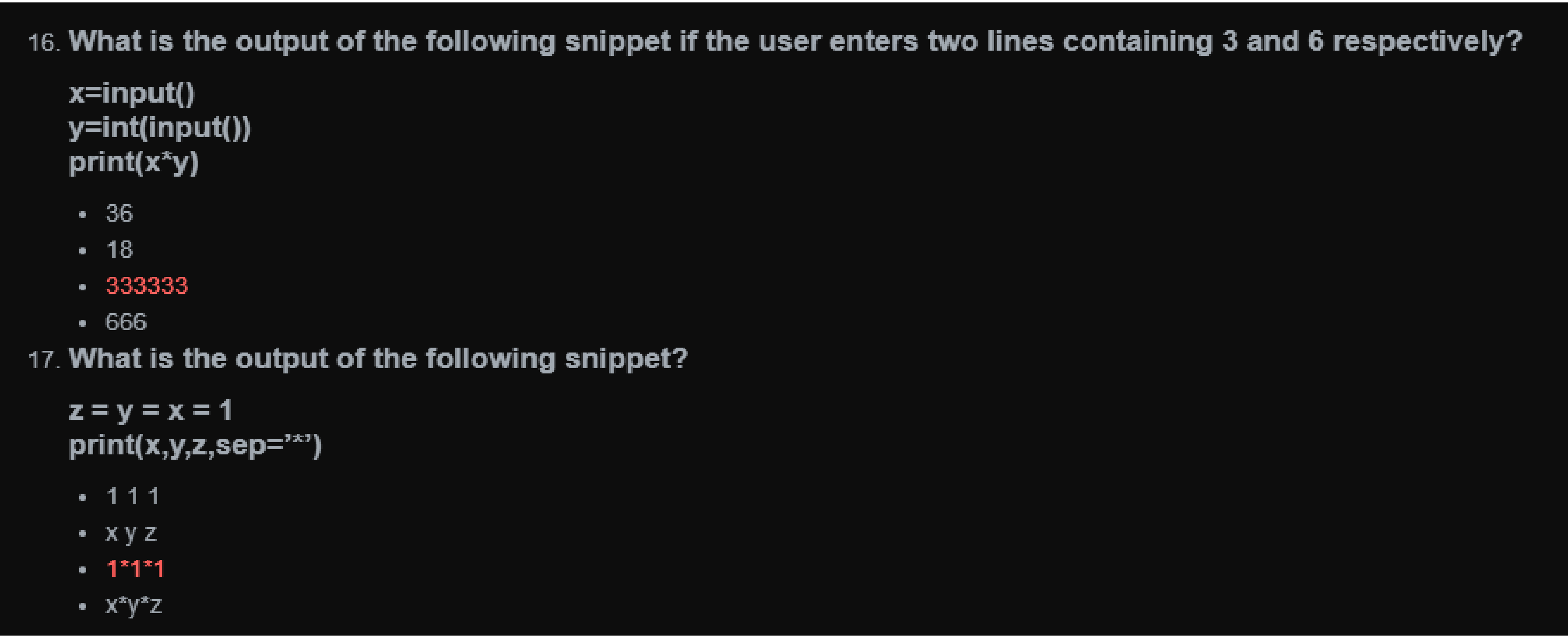
```
x=int(input())
y=int(input())
x=x/y
y=y/x
print(y)
```

- the code will cause a runtime error
- [✓ 8.0]
- 4.0
- 2.0

15. **What is the output of the following snippet if the user enters two lines containing 11 and 4 respectively?**

```
x=int(input())
y=int(input())
x = x % y
x = x % y
y = y % x
print(y)
```

- 3
- 2
- [✓ 1]
- 4



16. **What is the output of the following snippet if the user enters two lines containing 3 and 6 respectively?**

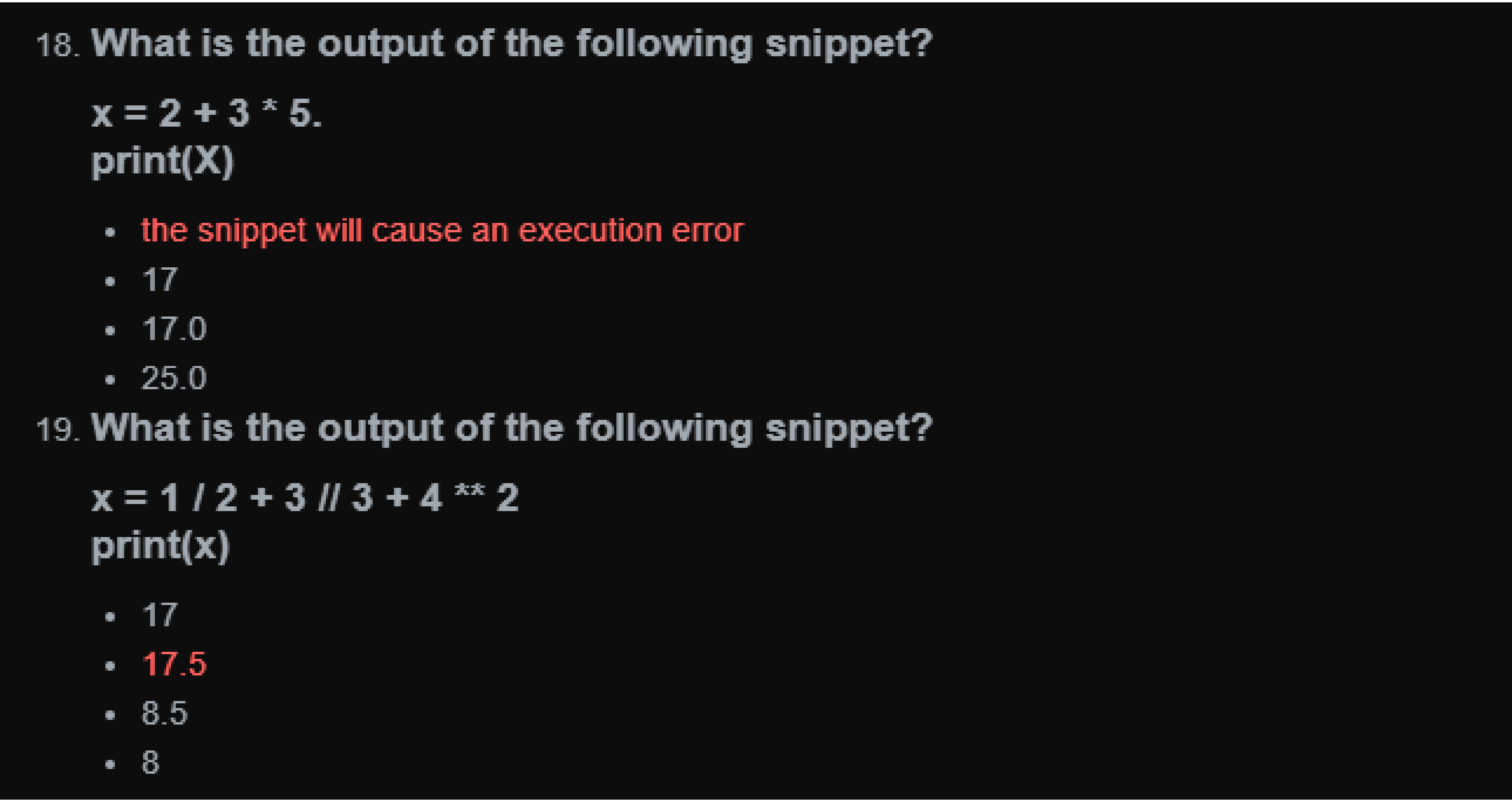
```
x=input()
y=int(input())
print(x*y)
```

- 36
- 18
- [✓ 333333]
- 666

17. **What is the output of the following snippet?**

```
z = y = x = 1
print(x,y,z,sep=’*’)
```

- 1 1 1
- x y z
- [✓ 1*1*1]
- x*y*z



18. **What is the output of the following snippet?**

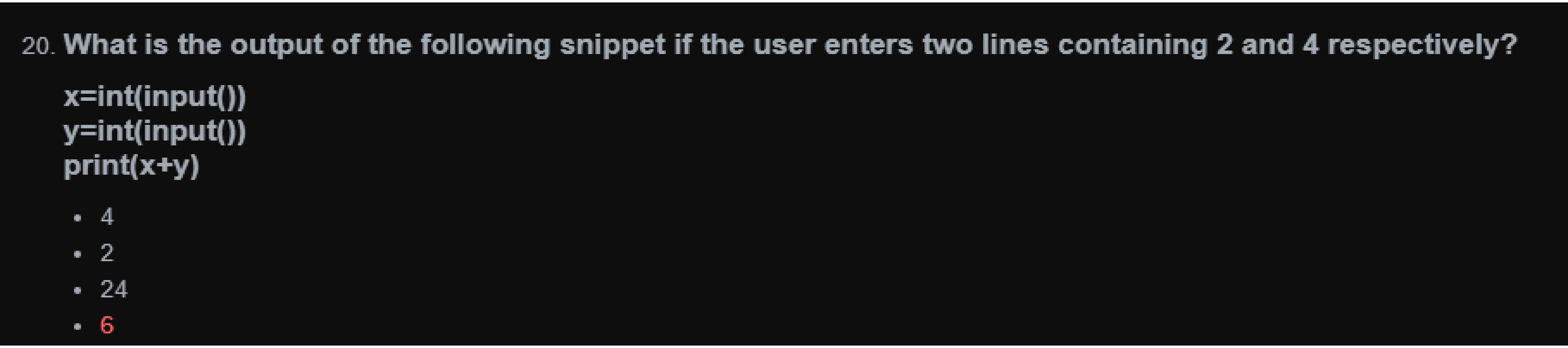
```
x = 2 + 3 * 5.
print(X)
```

- [✓ the snippet will cause an execution error]
- 17
- 17.0
- 25.0

19. **What is the output of the following snippet?**

```
x = 1 / 2 + 3 // 3 + 4 ** 2
print(x)
```

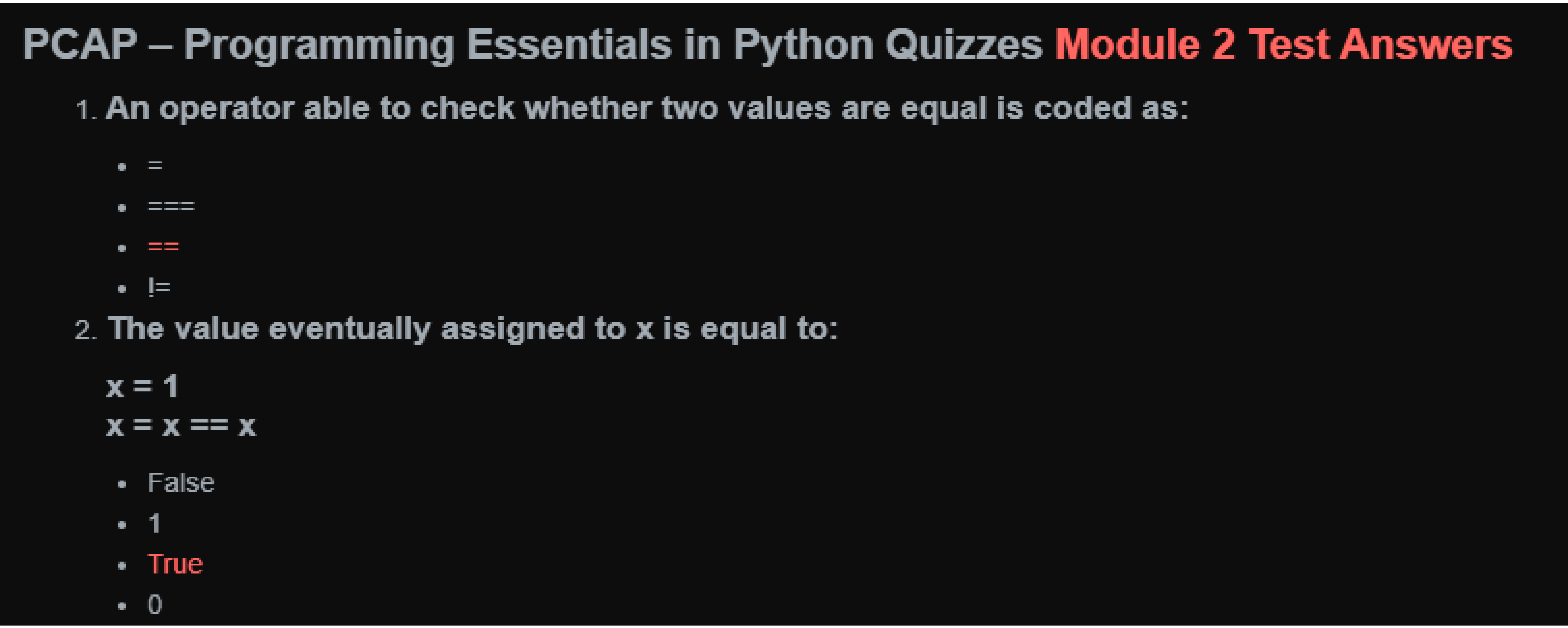
- 17
- [✓ 17.5]
- 8.5
- 8



20. **What is the output of the following snippet if the user enters two lines containing 2 and 4 respectively?**

```
x=int(input())
y=int(input())
print(x+y)
```

- 4
- 2
- 24
- [✓ 6]



PCAP – Programming Essentials in Python Quizzes Module 2 Test Answers

1. **An operator able to check whether two values are equal is coded as:**

   - =
   - ===
   - [✓ ==]
   - !=

2. **The value eventually assigned to x is equal to:**

   ```
   x = 1
   x = x == x
   ```

   - False
   - 1
   - [✓ True]
   - 0

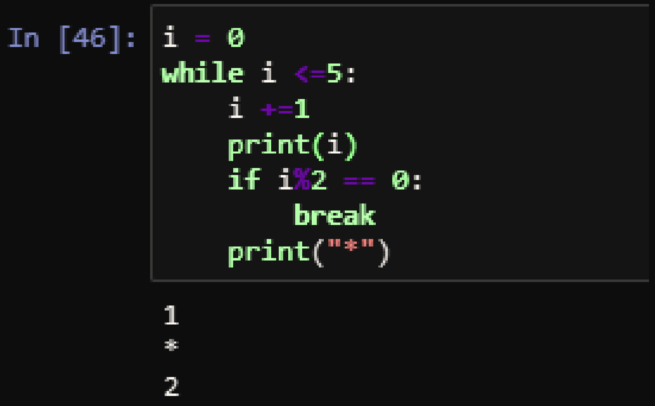
3. **How many stars will the following snippet send to the console?**

```
i = 0
while i <= 3 :
i += 2
print(“*”)
```

- two
- zero
- one
- three

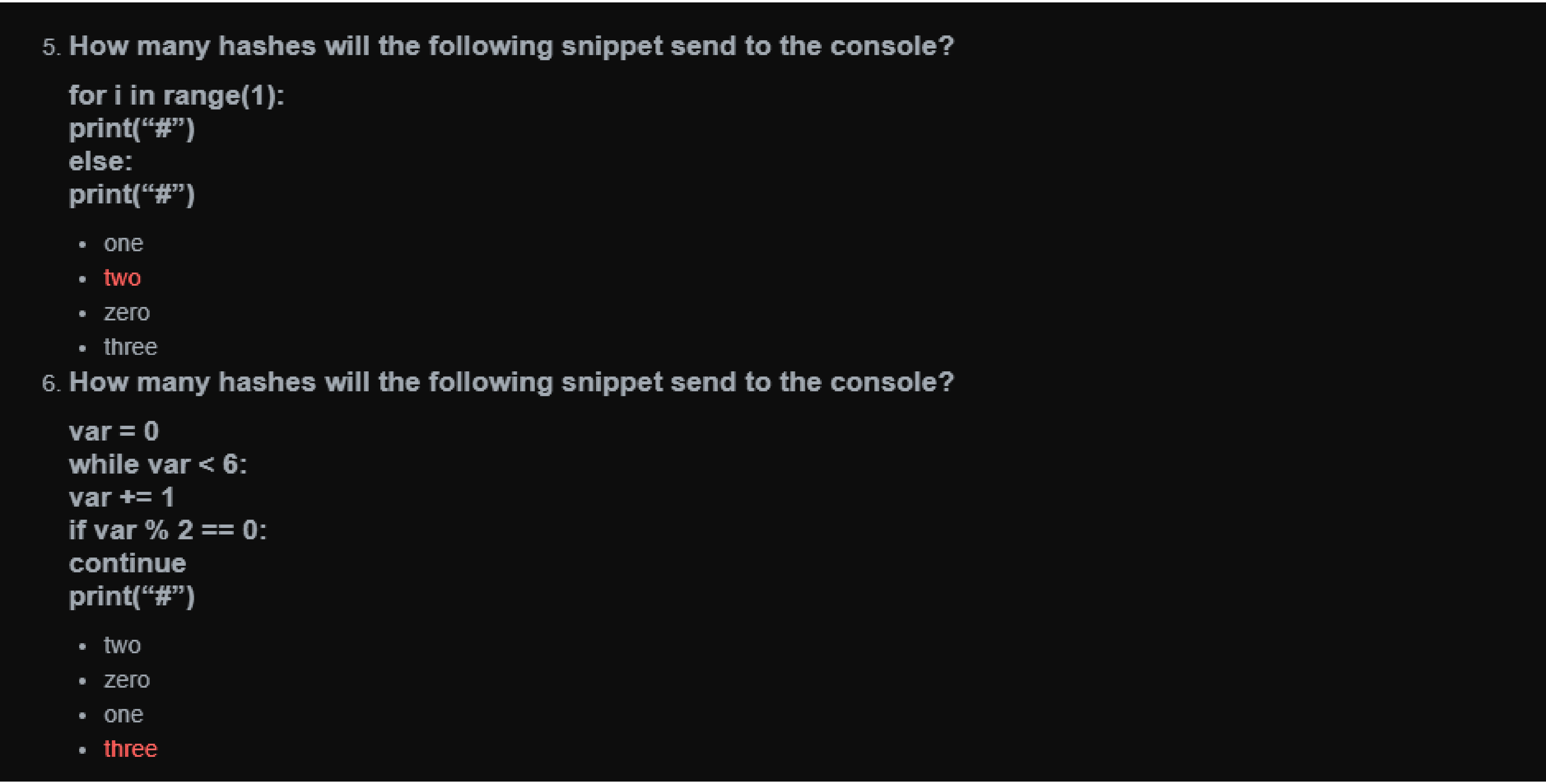
4. **How many stars will the following snippet send to the console?**

```
i = 0
while i <= 5 :
i += 1
if i % 2 == 0:
break
print(“*”)
```



```
In [46]: i = 0
         while i <=5:
             i +=1
             print(i)
             if i%2 == 0:
                 break
             print("*")

         1
         *
         2
```

- zero
- one
- two
- three



5. **How many hashes will the following snippet send to the console?**

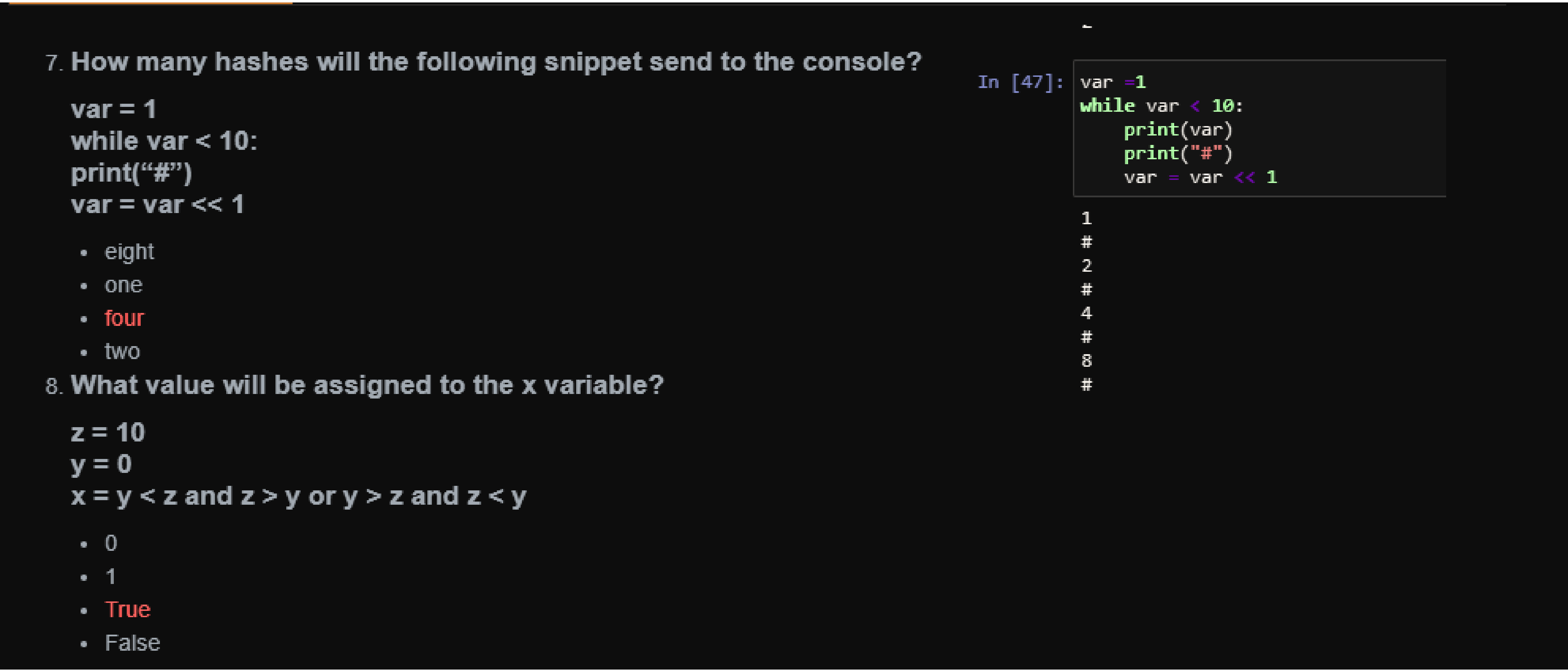
```
for i in range(1):
print("#")
else:
print("#")
```

- one
- [✓ two]
- zero
- three

6. **How many hashes will the following snippet send to the console?**

```
var = 0
while var < 6:
var += 1
if var % 2 == 0:
continue
print("#")
```

- two
- zero
- one
- [✓ three]



7. **How many hashes will the following snippet send to the console?**

```
var = 1
while var < 10:
print(“#”)
var = var << 1
```

- eight
- one
- [✓ four]
- two

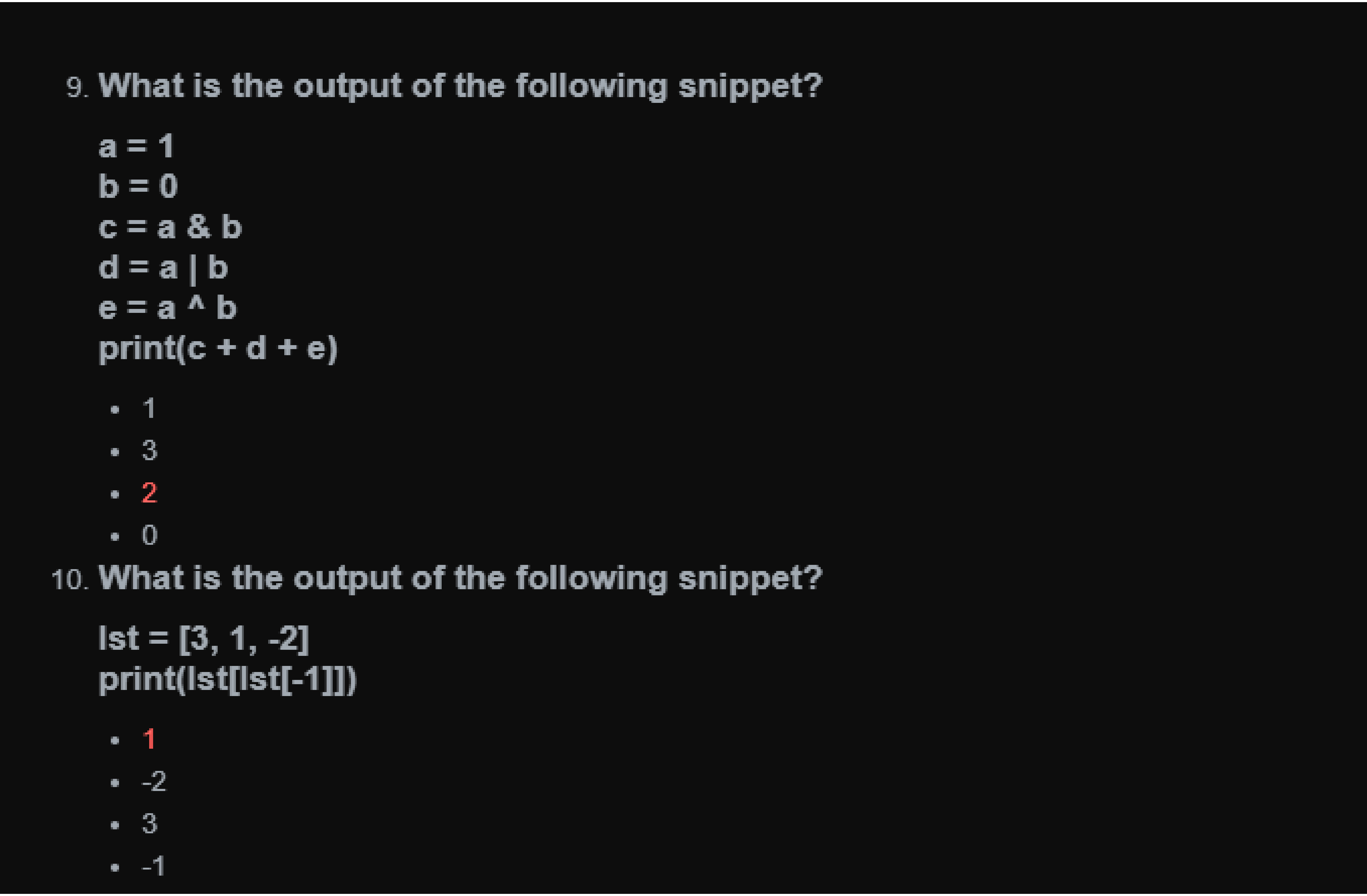
8. **What value will be assigned to the x variable?**

```
z = 10
y = 0
x = y < z and z > y or y > z and z < y
```

- 0
- 1
- [✓ True]
- False

```
In [47]: var =1
         while var < 10:
             print(var)
             print("#")
             var = var << 1
```

```
1
#
2
#
4
#
8
#
```



9. **What is the output of the following snippet?**

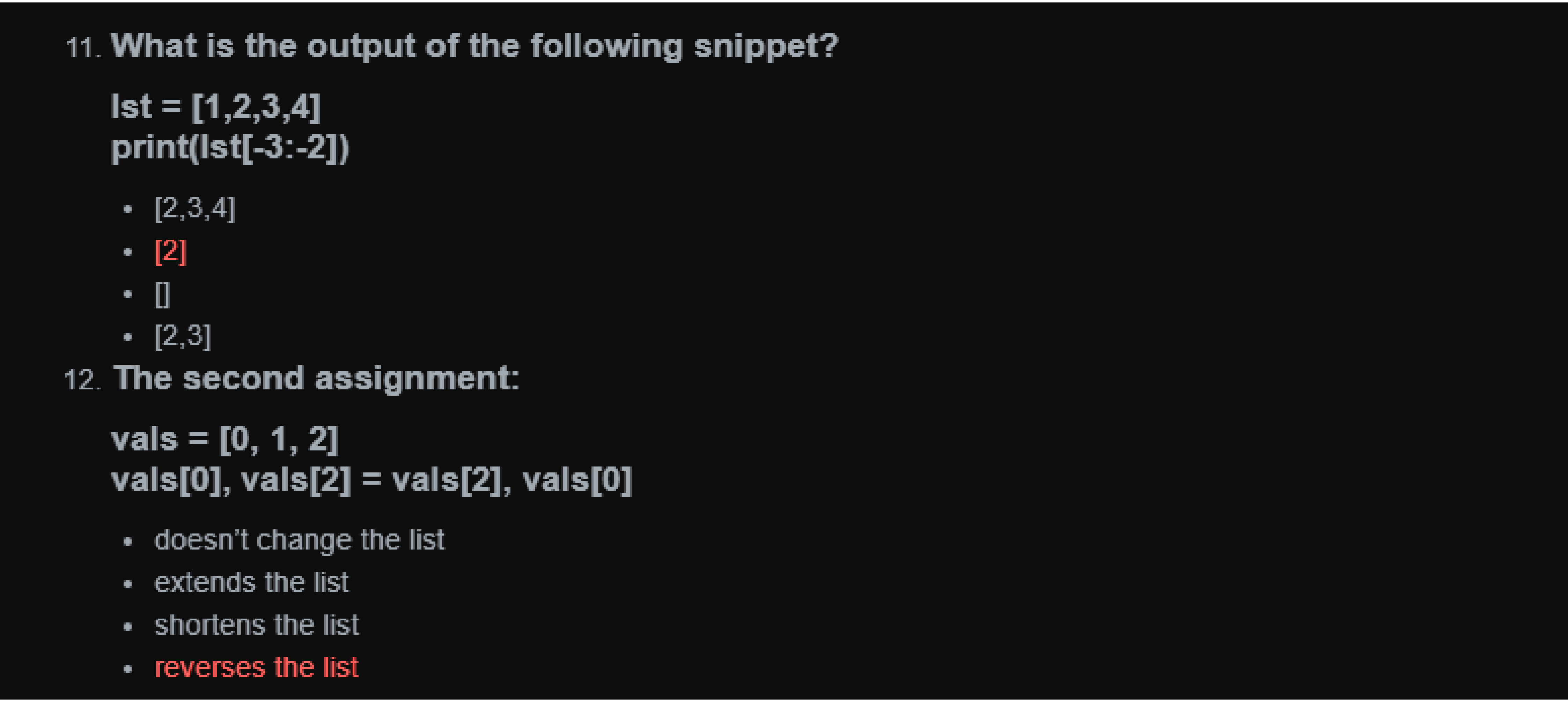
```
a = 1
b = 0
c = a & b
d = a | b
e = a ^ b
print(c + d + e)
```

- 1
- 3
- [✓ 2]
- 0

10. **What is the output of the following snippet?**

```
lst = [3, 1, -2]
print(lst[lst[-1]])
```

- [✓ 1]
- -2
- 3
- -1



11. **What is the output of the following snippet?**

```
lst = [1,2,3,4]
print(lst[-3:-2])
```

- [2,3,4]
- [✓ [2]]
- []
- [2,3]

12. **The second assignment:**

```
vals = [0, 1, 2]
vals[0], vals[2] = vals[2], vals[0]
```

- doesn’t change the list
- extends the list
- shortens the list
- [✓ reverses the list]

13. **After execution of the following snippet, the sum of all vals elements will be equal to:**

```
vals = [0, 1, 2]
vals.insert(0,1)
del vals[1]
```

- 2
- 5
- 3
- [✓ 4]

14. **Take a look at the snippet, and choose the true statement:**

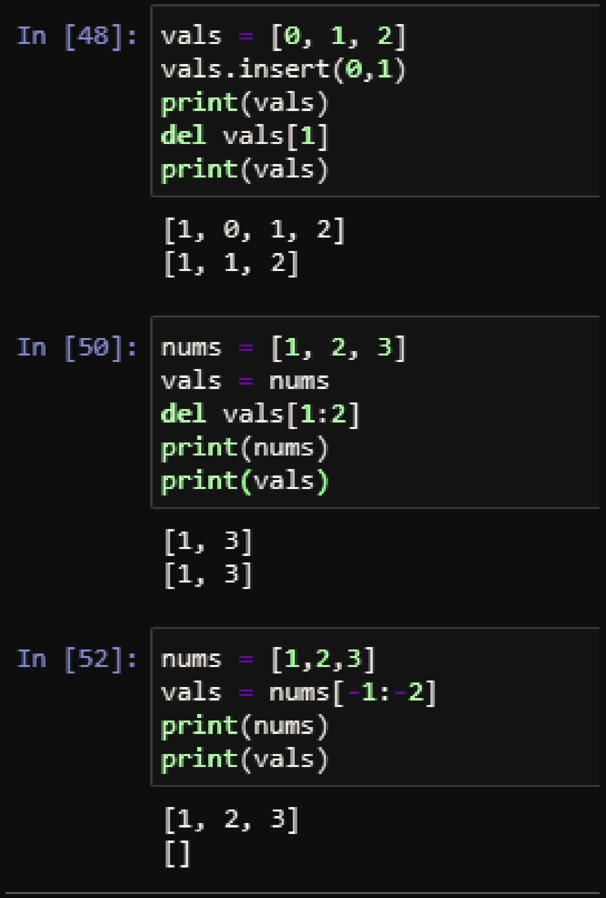
```
nums = [1,2,3]
vals = nums
del vals[1:2]
```

- nums is longer than vals
- vals is longer than nums
- [✓ nums and vals are of the same length]
- the snippet will cause a runtime error

15. **Which of the following sentences is true?**

```
nums = [1,2,3]
vals = nums[-1:-2]
```

- [✓ nums is longer than vals]
- nums and vals are of the same length
- the snippet will cause a runtime error
- vals is longer than nums



```
In [48]: vals = [0, 1, 2]
         vals.insert(0,1)
         print(vals)
         del vals[1]
         print(vals)

         [1, 0, 1, 2]
         [1, 1, 2]

In [50]: nums = [1, 2, 3]
         vals = nums
         del vals[1:2]
         print(nums)
         print(vals)

         [1, 3]
         [1, 3]

In [52]: nums = [1,2,3]
         vals = nums[-1:-2]
         print(nums)
         print(vals)

         [1, 2, 3]
         []
```

16. **What is the output of the following snippet?**

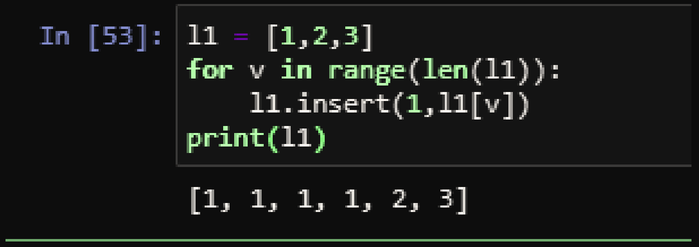
```
l1 = [1,2,3]
l2 = []
for v in l1:
l2.insert(0,v)
print(l2)
```

- [✓ [3,2,1]]
- [1,2,3]
- [3,3,3]
- [1,1,1]

17. **What is the output of the following snippet?**

```
l1 = [1,2,3]
for v in range(len(l1)):
l1.insert(1,l1[v])
print(l1)
```

- [1, 2, 3, 3, 2, 1]
- [1, 2, 3, 1, 2, 3]
- [3, 2, 1, 1, 2, 3]
- [✓ [1, 1, 1, 1, 2, 3]]



```
In [53]: l1 = [1,2,3]
         for v in range(len(l1)):
             l1.insert(1,l1[v])
         print(l1)

         [1, 1, 1, 1, 2, 3]
```

18. **How many elements does the L list contain?**

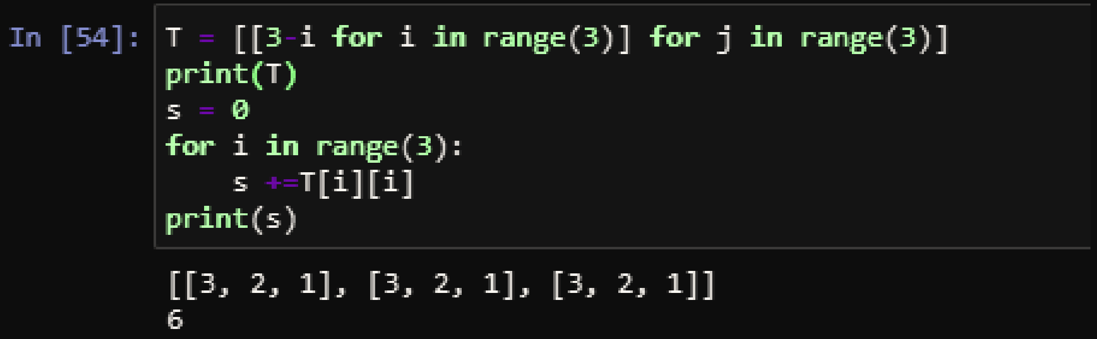
L = [i for i in range(-1,2)]

- one
- four
- [✓ three]
- two

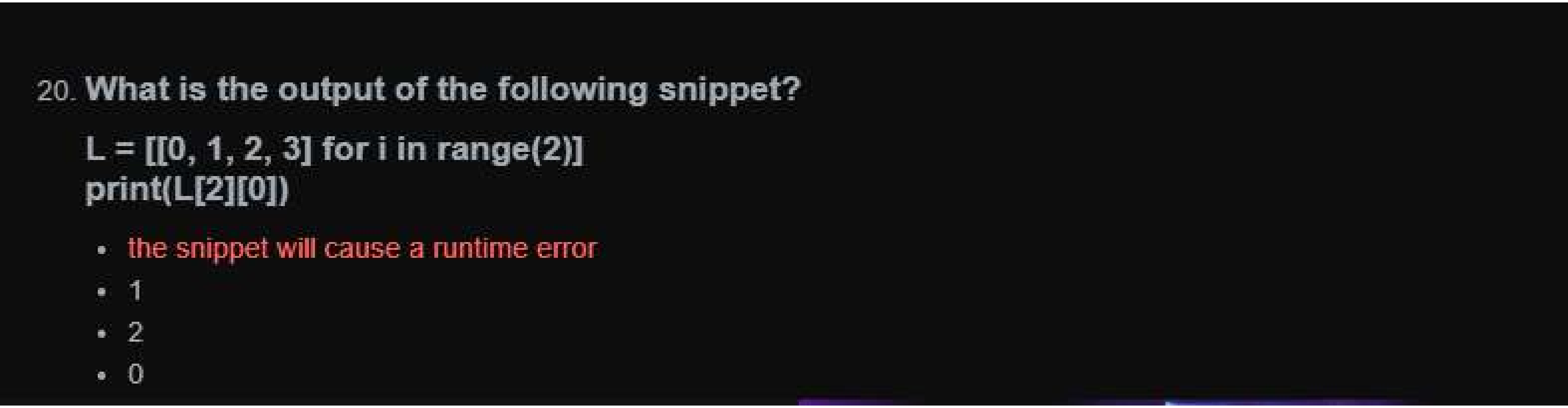
19. **What is the output of the following snippet?**

T = [[3-i for i in range (3)] for j in range (3)]
s = 0
for i in range(3):
s += T[i][i]
print(s)

- 4
- 2
- 7
- [✓ 6]



```
In [54]: T = [[3-i for i in range(3)] for j in range(3)]
         print(T)
         s = 0
         for i in range(3):
             s +=T[i][i]
         print(s)

[[3, 2, 1], [3, 2, 1], [3, 2, 1]]
6
```



20. **What is the output of the following snippet?**

```
L = [[0, 1, 2, 3] for i in range(2)]
print(L[2][0])
```

- the snippet will cause a runtime error
- 1
- 2
- 0

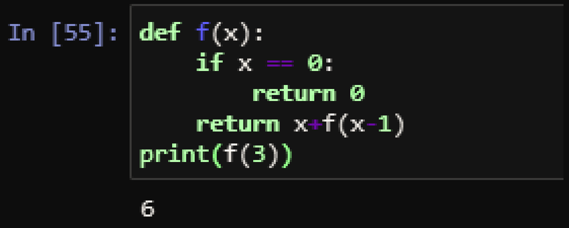
PCAP – Programming Essentials in Python Quizzes Module 3 Test Answers

1. **Which of the following lines properly starts a parameterless function definition?**
   - function fun():
   - def fun:
   - fun function():
   - [✓ def fun():]
2. **A function defined in the following way:**

   ```
   def function(x=0):
   return x
   ```

   - [✓ may be invoked without any argument, or with just one]
   - must be invoked without arguments
   - must be invoked with exactly one argument
   - may be invoked with any number of arguments (including zero)
3. **A built-in function is a function which:**
   - has to be imported before use
   - is hidden from programmers
   - has been placed within your code by another programmer
   - [✓ comes with Python, and is an integral part of Python]
4. **The fact that tuples belong to sequence types means:**
   - they can be modified using the del instruction
   - [✓ they can be indexed and sliced like lists]
   - they are actually lists
   - they can be extended using the .append() method
5. **What is the output of the following snippet?**

   ```
   def f(x):
   if x == 0:
   return 0
   return x + f(x – 1)

   print(f(3))
   ```



```
In [55]: def f(x):
             if x == 0:
                 return 0
             return x+f(x-1)
         print(f(3))

         6
```

   - the code is erroneous
   - 3
   - 1
   - [✓ 6]

6. **What is the output of the following snippet?**

```
def fun(x):
x += 1
return x
```

```
x = 2
x = fun(x+1)
print(x)
```

- the code is erroneous
- [✓ 4]
- 5
- 3

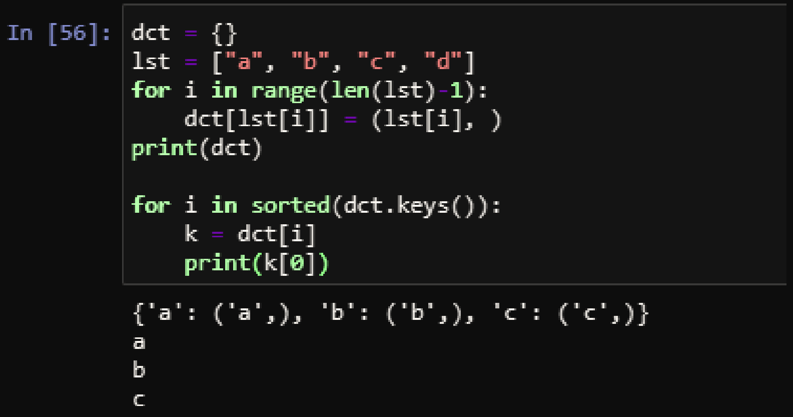
7. **What code would you insert into the commented line to obtain the output that reads:**

```
a
b
c
```

Code:

```
dct = { }
lst = [‘a’,’b’,’c’,’d’]
for i in range(len(lst) – 1):
dct[lst[i]] = ( lst[i], )
for i in sorted(dct.keys()):
k = dct[i]
# insert your code
```

- [✓ print(k[0])]
- print(k)
- print(k[“0”])
- print(k[‘0’])



```
In [56]: dct = {}
         lst = ["a", "b", "c", "d"]
         for i in range(len(lst)-1):
             dct[lst[i]] = (lst[i], )
         print(dct)

         for i in sorted(dct.keys()):
             k = dct[i]
             print(k[0])

         {'a': ('a',), 'b': ('b',), 'c': ('c',)}
         a
         b
         c
```

8. **The following snippet:**

```
def func(a,b):
return a ** a
```

```
print(func(2))
```

- will output 2
- [✓ is erroneous]
- will return None
- will output 4

9. **The following snippet:**

```
def func1(a):
return a ** a
def func2(a):
return func1(a)*func1(a)
print(func2(2))
```

- is erroneous
- will output 2
- will output 4
- [✓ will output 16]

10. **Which of the following lines properly starts a function using two parameters, both with zeroed default values?**

- def fun(a=b=0):
- fun fun(a,b=0):
- fun fun(a=0,b):
- [✓ def fun(a=0,b=0):]

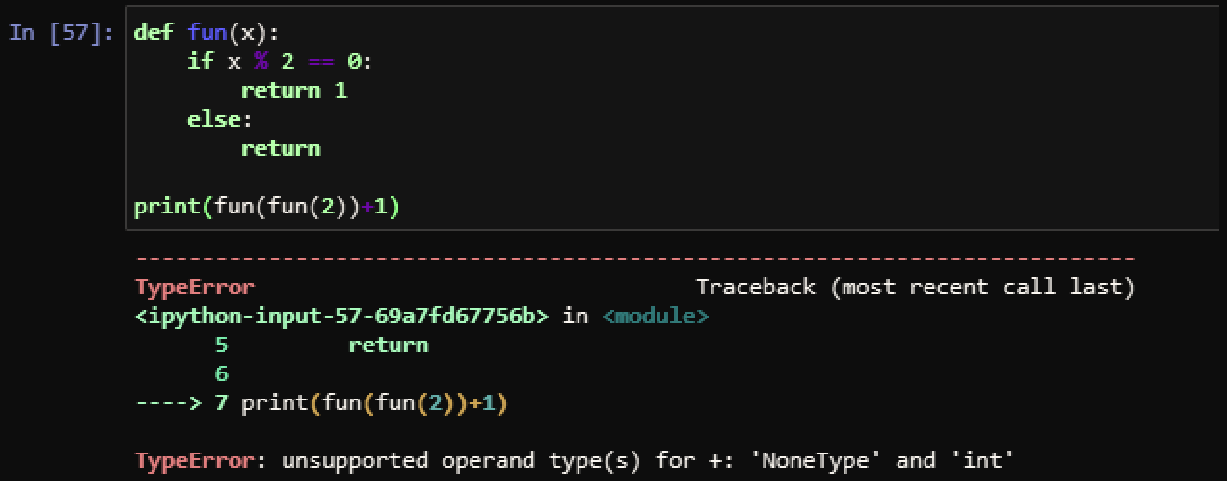
11. **Which of the following statements is false?**

- The None value cannot be used as an argument of arithmetic operators
- The None value can be assigned to variables
- [✓ The None value may not be used outside functions]
- The None value can be compared with variables

12. **What is the output of the following snippet?**

```
def fun(x):
if x % 2 == 0:
return 1
else:
return

print(fun(fun(2)) + 1)
```

- [✓ the code will cause a runtime error]
- 1
- None
- 2



```
In [57]: def fun(x):
             if x % 2 == 0:
                 return 1
             else:
                 return

         print(fun(fun(2))+1)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-57-69a7fd67756b> in <module>
      5         return
      6
----> 7 print(fun(fun(2))+1)

TypeError: unsupported operand type(s) for +: 'NoneType' and 'int'
```

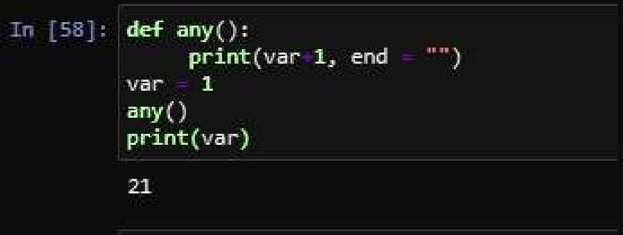
13. **What is the output of the following snippet?**

```
def fun(x):
global y
y = x * x
return y

fun(2)
print(y)
```

- 2
- [✓ 4]
- None
- the code will cause a runtime error

14. **What is the output of the following snippet?**

```
def any():
print(var + 1,end=”)
var = 1
any()
print(var)
```



```
In [58]: def any():
              print(var+1, end = "")
         var = 1
         any()
         print(var)

         21
```

- 22
- [✓ 21]
- 11
- 12

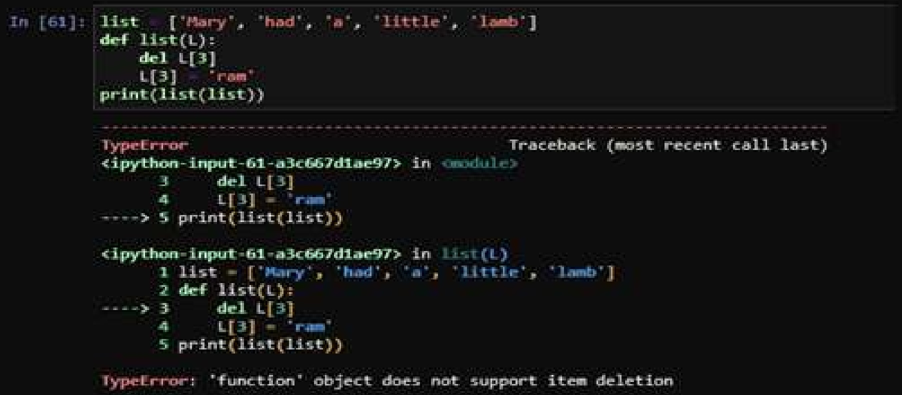
15. **Assuming that tuple is a correctly created tuple, the fact that tuples are immutable means that the following instruction:**

```
tuple[1] = tuple[1] + tuple[0]
```

- [✓ is illegal]
- is fully correct
- can be executed if and only if the tuple contains at least two elements
- may be illegal if the tuple contains strings

16. **What is the output of the following snippet?**

```
list = [‘Mary’, ‘had’, ‘a’, ‘little’, ‘lamb’]
def list(L):
del L[3]
L[3] = ‘ram’
print(list(list))
```



```
In [61]: list = ['Mary', 'had', 'a', 'little', 'lamb']
         def list(L):
             del L[3]
             L[3] = 'ram'
         print(list(list))

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-61-a3c667d1ae97> in <module>
      3     del L[3]
      4     L[3] = 'ram'
----> 5 print(list(list))

<ipython-input-61-a3c667d1ae97> in list(L)
      1 list = ['Mary', 'had', 'a', 'little', 'lamb']
      2 def list(L):
----> 3     del L[3]
      4     L[3] = 'ram'
      5 print(list(list))

TypeError: 'function' object does not support item deletion
```

- [‘Mary’, ‘had’, ‘a’, ‘ram’]
- [‘Mary’, ‘had’, ‘a’, ‘little’, ‘lamb’]
- [✓ the snippet is erroneous]
- [‘Mary’, ‘had’, ‘a’, ‘lamb’]

17. **What is the output of the following snippet?**

17. **What is the output of the following snippet?**

```
def fun(x,y,z):
return x+2*y+3*z

print(fun(0,z=1,y=3))
```

- 3
- the snippet is erroneous
- 0
- 9

18. **What is the output of the following snippet?**

```
def fun(inp=2,out=3):
return inp * out
print(fun(out=2))
```

- 6
- 4
- 2
- the snippet is erroneous

19. **What is the output of the following snippet?**

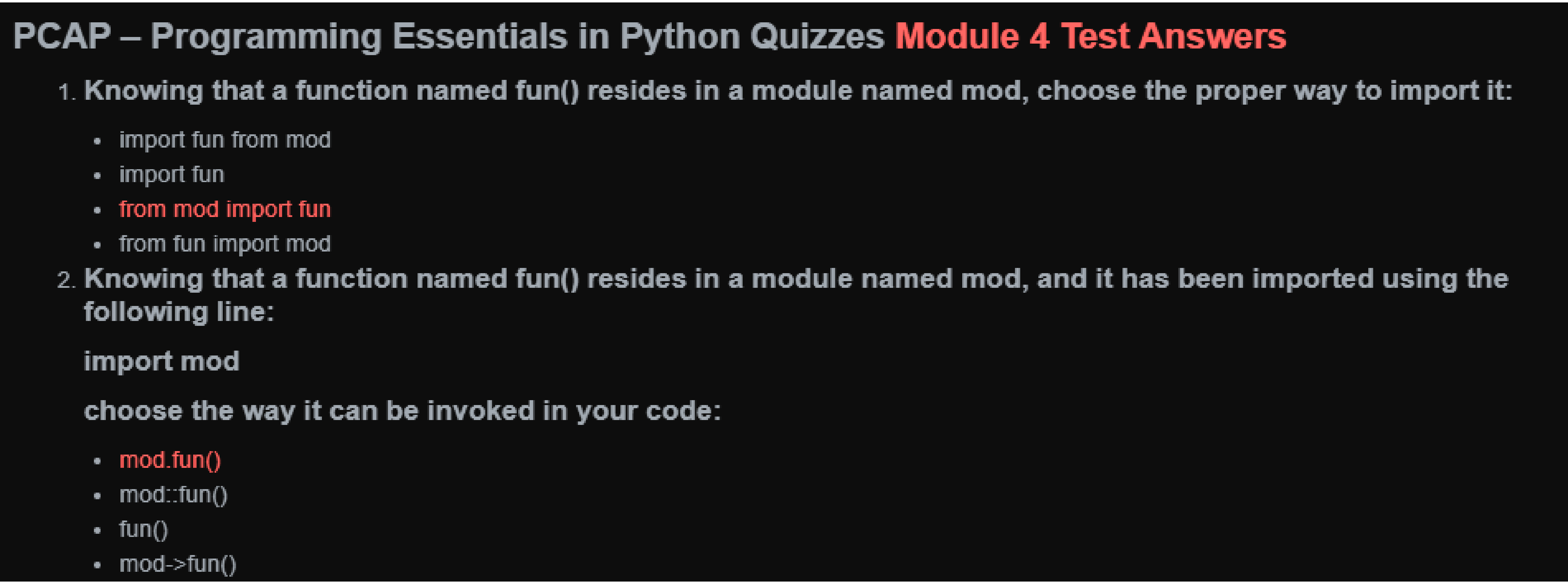
```
dct = { ‘one’:’two’, ‘three’:’one’, ‘two’:’three’ }
v = dct[‘one’]
for k in range(len(dct)):
v = dct[v]
print(v)
```

- two
- three
- (‘one’, ‘two’, ‘three’)
- one

20. **What is the output of the following snippet?**

```
tup = (1, 2, 4, 8)
tup = tup[1:-1]
tup = tup[0]
print(tup)
```

- the snippet is erroneous
- (2)
- (2,)
- 2

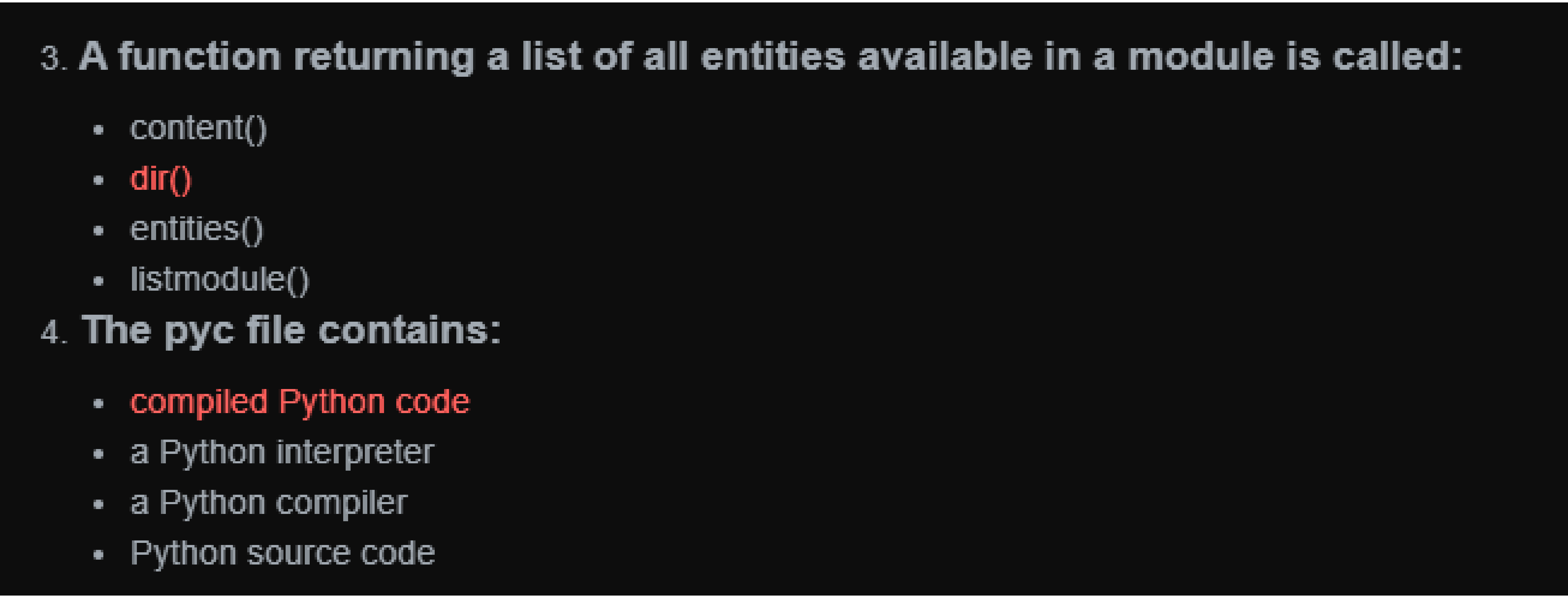


PCAP – Programming Essentials in Python Quizzes Module 4 Test Answers

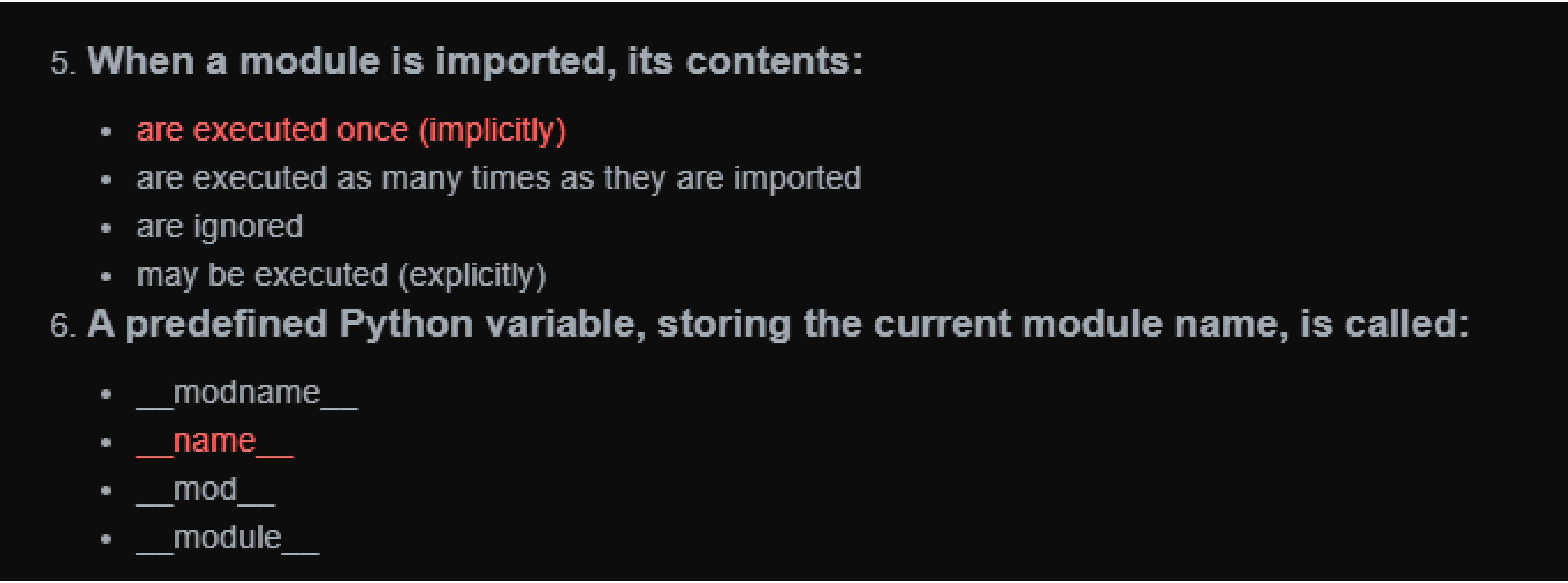
1. **Knowing that a function named fun() resides in a module named mod, choose the proper way to import it:**
   - import fun from mod
   - import fun
   - [✓ from mod import fun]
   - from fun import mod
2. **Knowing that a function named fun() resides in a module named mod, and it has been imported using the following line:**

   import mod

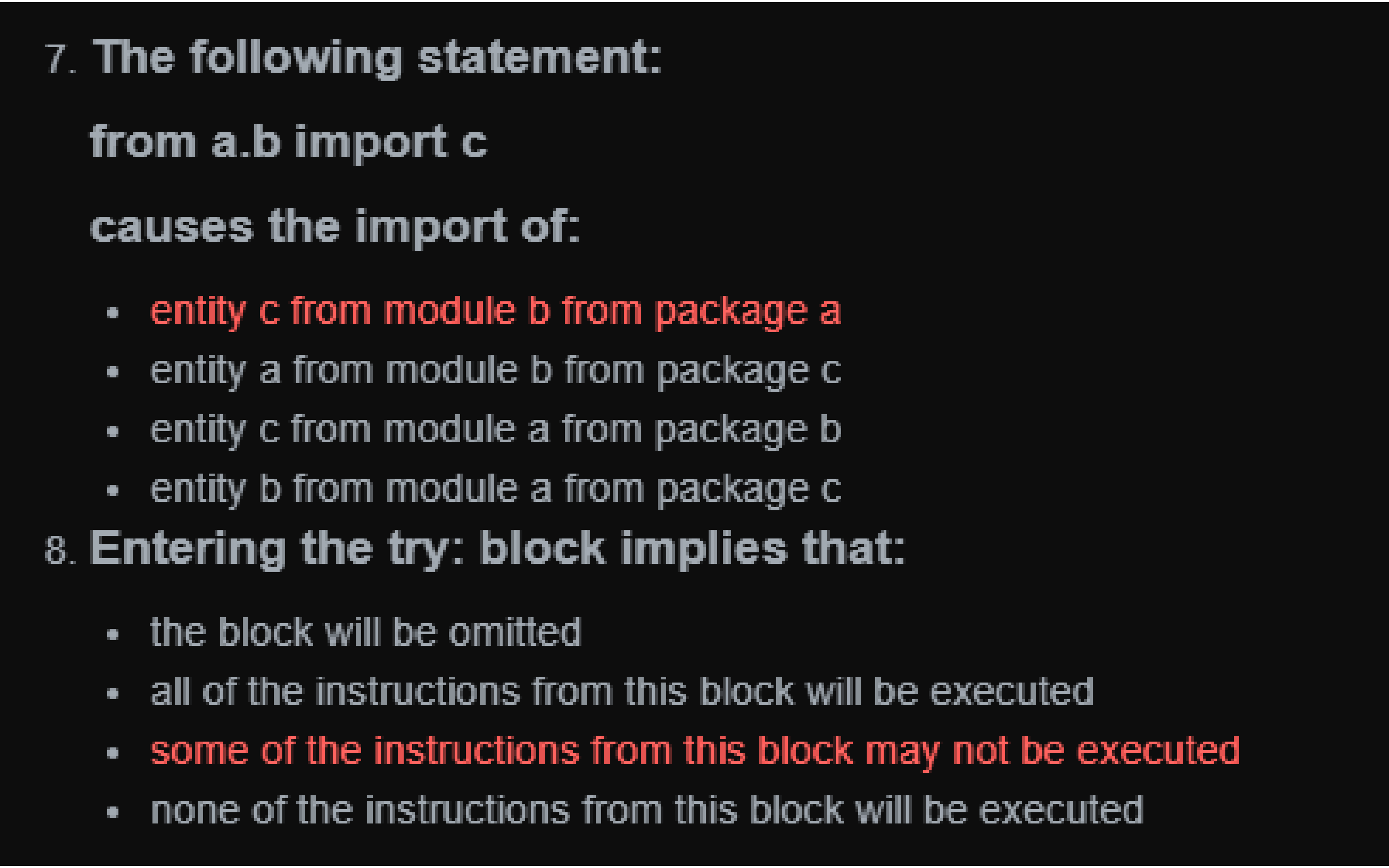
   choose the way it can be invoked in your code:
   - [✓ mod.fun()]
   - mod::fun()
   - fun()
   - mod->fun()



3. **A function returning a list of all entities available in a module is called:**
   - content()
   - [✓ dir()]
   - entities()
   - listmodule()
4. **The pyc file contains:**
   - [✓ compiled Python code]
   - a Python interpreter
   - a Python compiler
   - Python source code



5. **When a module is imported, its contents:**
   - [✓ are executed once (implicitly)]
   - are executed as many times as they are imported
   - are ignored
   - may be executed (explicitly)
6. **A predefined Python variable, storing the current module name, is called:**
   - __modname__
   - [✓ __name__]
   - __mod__
   - __module__



7. **The following statement:**

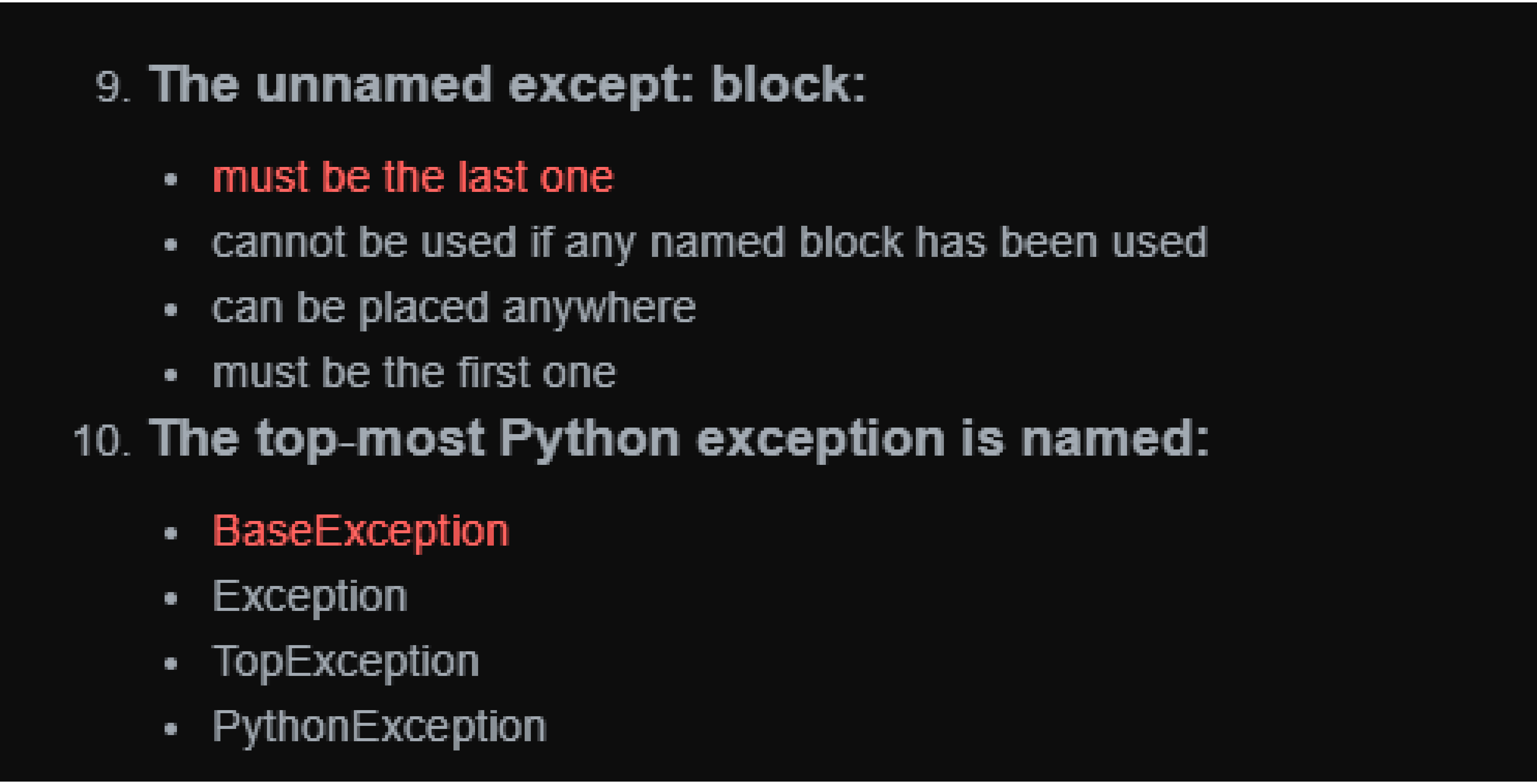
   from a.b import c

   causes the import of:

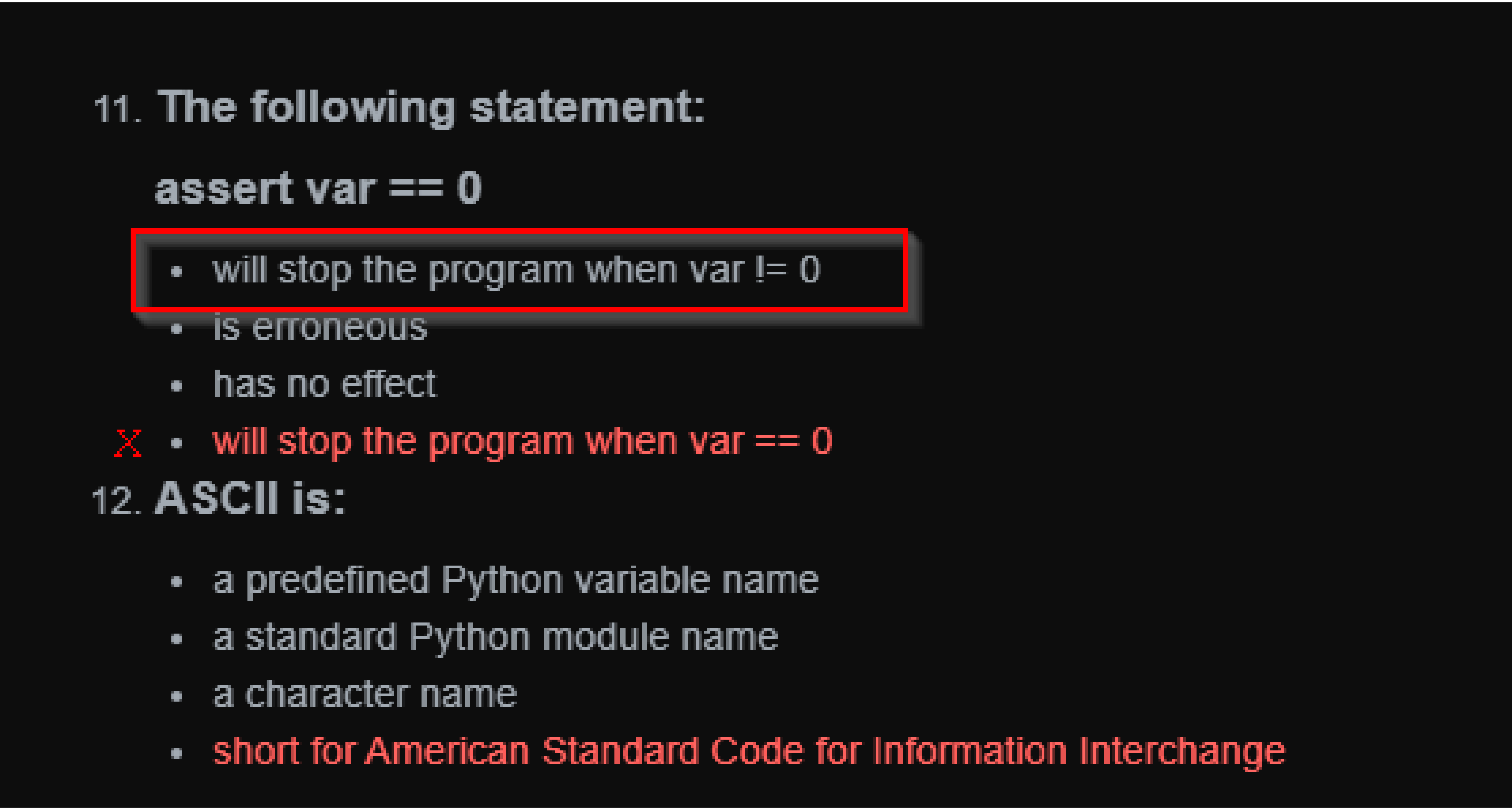
   - entity c from module b from package a
   - entity a from module b from package c
   - entity c from module a from package b
   - entity b from module a from package c

8. **Entering the try: block implies that:**

   - the block will be omitted
   - all of the instructions from this block will be executed
   - some of the instructions from this block may not be executed
   - none of the instructions from this block will be executed



9. **The unnamed except: block:**
    - [✓ must be the last one]
    - cannot be used if any named block has been used
    - can be placed anywhere
    - must be the first one
10. **The top-most Python exception is named:**
    - [✓ BaseException]
    - Exception
    - TopException
    - PythonException



11. **The following statement:**

    assert var == 0

    - [✓ will stop the program when var != 0]
    - is erroneous
    - has no effect
    - ✗ will stop the program when var == 0

12. **ASCII is:**

    - a predefined Python variable name
    - a standard Python module name
    - a character name
    - short for American Standard Code for Information Interchange

13. **UTF-8 is:**

- a synonym for “byte”
- a form of encoding Unicode code points
- the 9th version of the UTF standard
- a Python version name

14. **UNICODE is a standard:**

- honored by the whole universe
- for coding floating-point numbers
- used by coders from universities
- like ASCII, but much more expansive

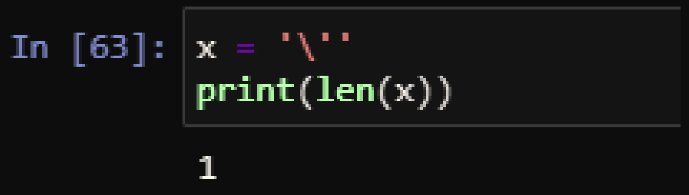
15. **The following code**

x = ‘\’”

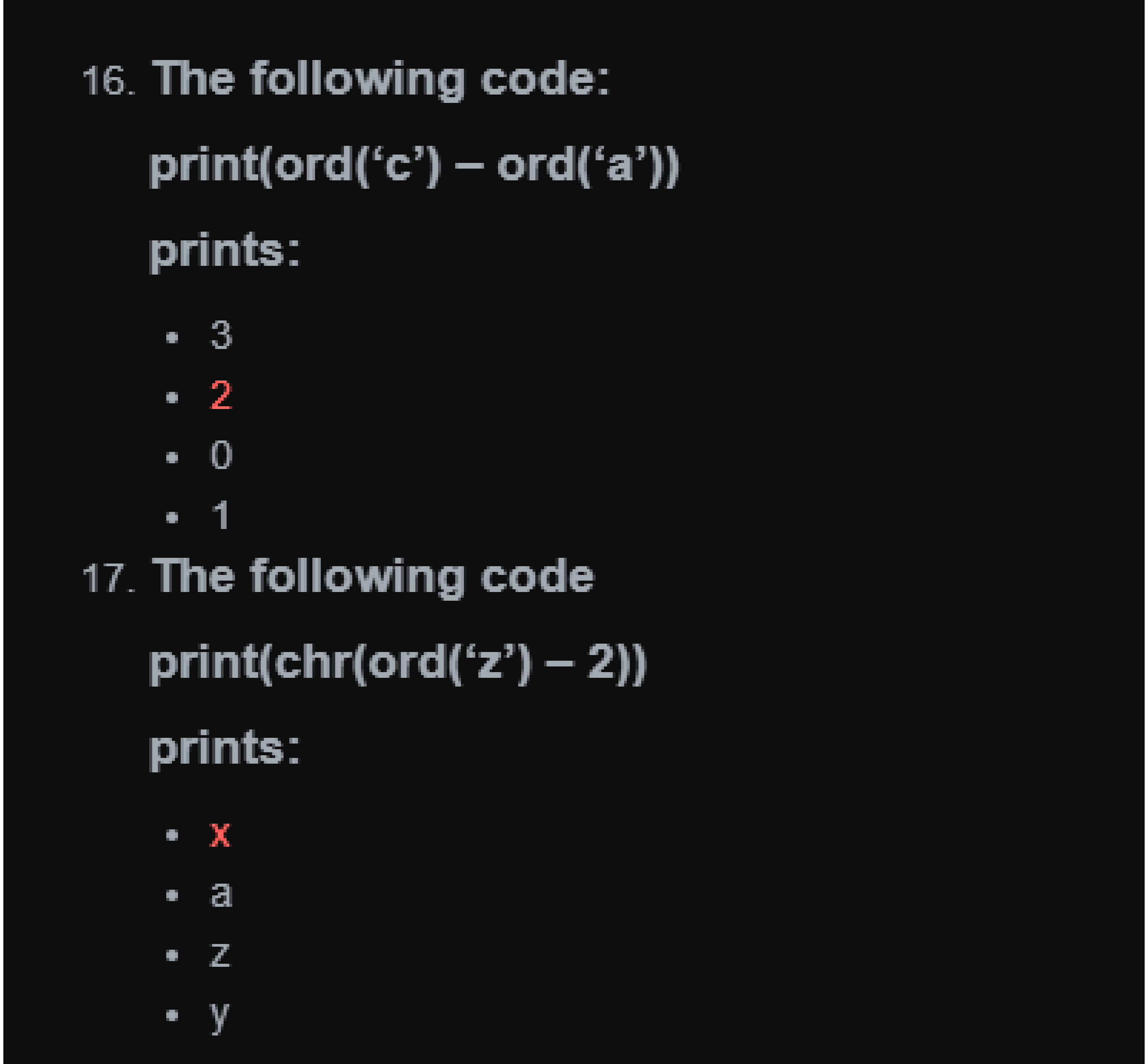
print(len(x))

prints:

- 1
- 0
- 3
- 2



```
In [63]: x = '\''
         print(len(x))

         1
```



16. **The following code:**

    print(ord(‘c’) – ord(‘a’))

    prints:

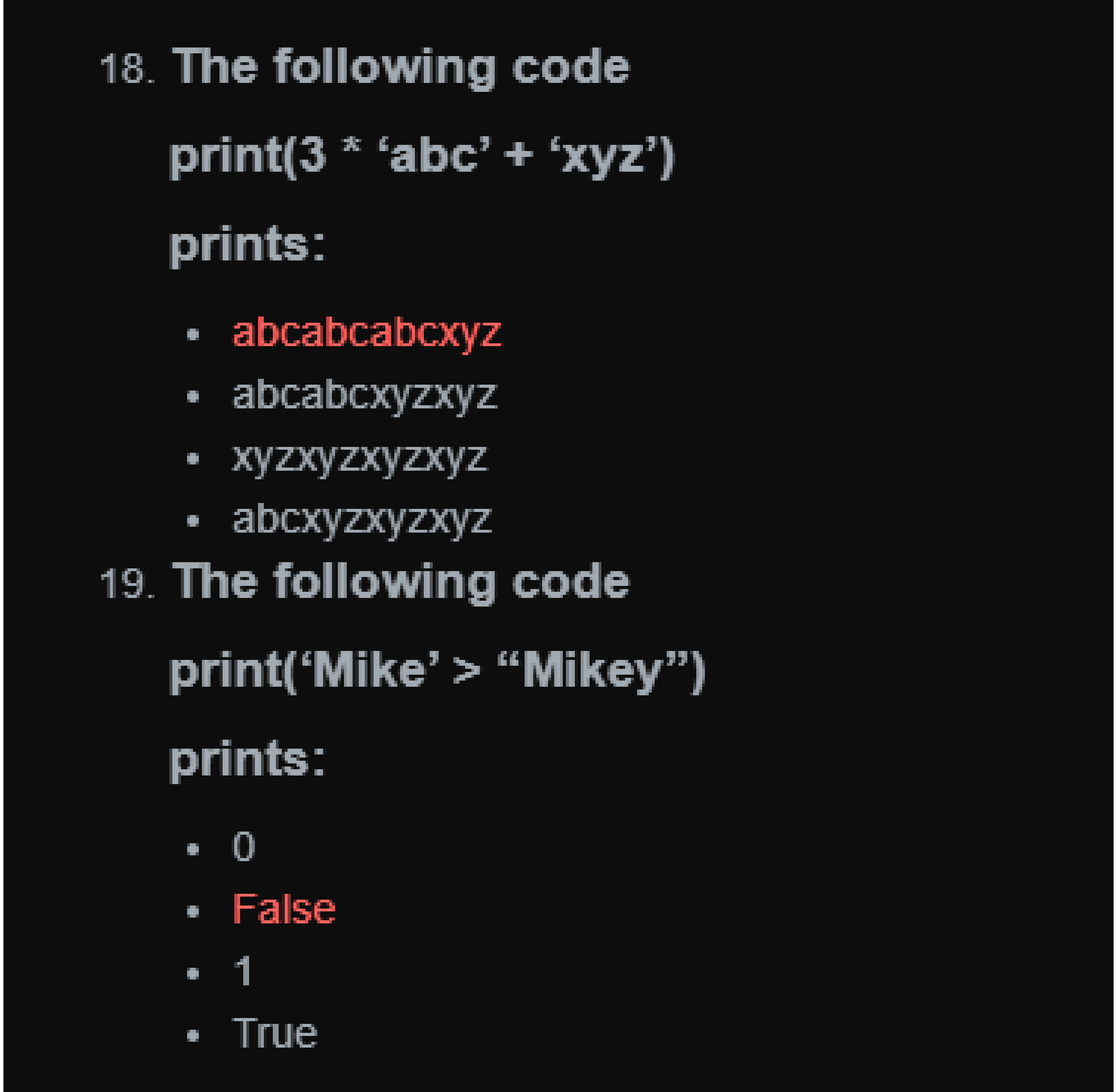
    - 3
    - 2
    - 0
    - 1

17. **The following code**

    print(chr(ord(‘z’) – 2))

    prints:

    - x
    - a
    - z
    - y



18. **The following code**

    ```
    print(3 * ‘abc’ + ‘xyz’)
    ```

    prints:

    - abcabcabcxyz
    - abcabcxyzxyz
    - xyzxyzxyzxyz
    - abcxyzxyzxyz

19. **The following code**

    ```
    print(‘Mike’ > “Mikey”)
    ```

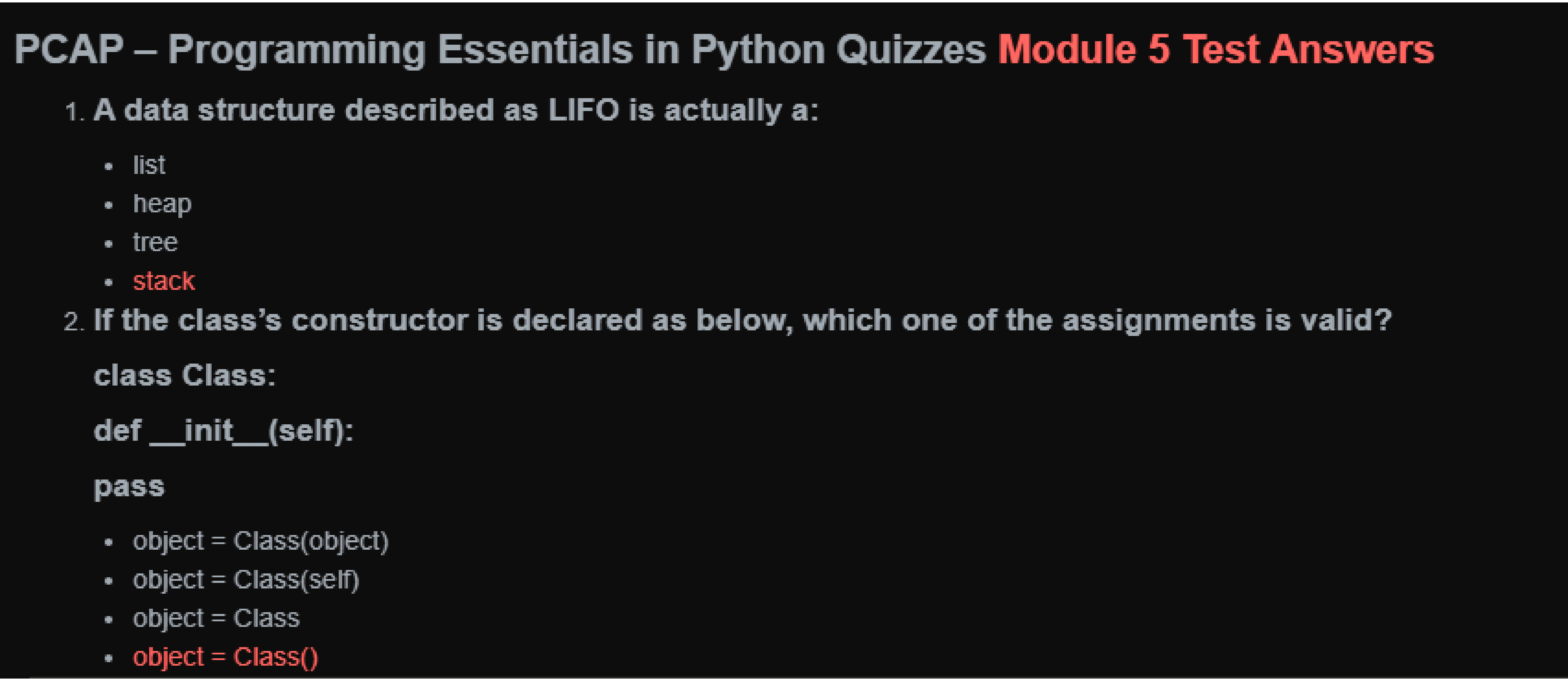
    prints:

    - 0
    - False
    - 1
    - True

20. **The following code:**

```
print(float("1,3"))
```

- prints 1,3
- prints 1.3
- [✓ raises a ValueError exception]
- prints 13



PCAP – Programming Essentials in Python Quizzes Module 5 Test Answers

1. **A data structure described as LIFO is actually a:**
   - list
   - heap
   - tree
   - [✓ stack]
2. **If the class's constructor is declared as below, which one of the assignments is valid?**

   ```
   class Class:
   def __init__(self):
   pass
   ```

   - object = Class(object)
   - object = Class(self)
   - object = Class
   - [✓ object = Class()]

3. **If there is a superclass named A and a subclass named B, which one of the presented invocations should you put instead of a comment?**

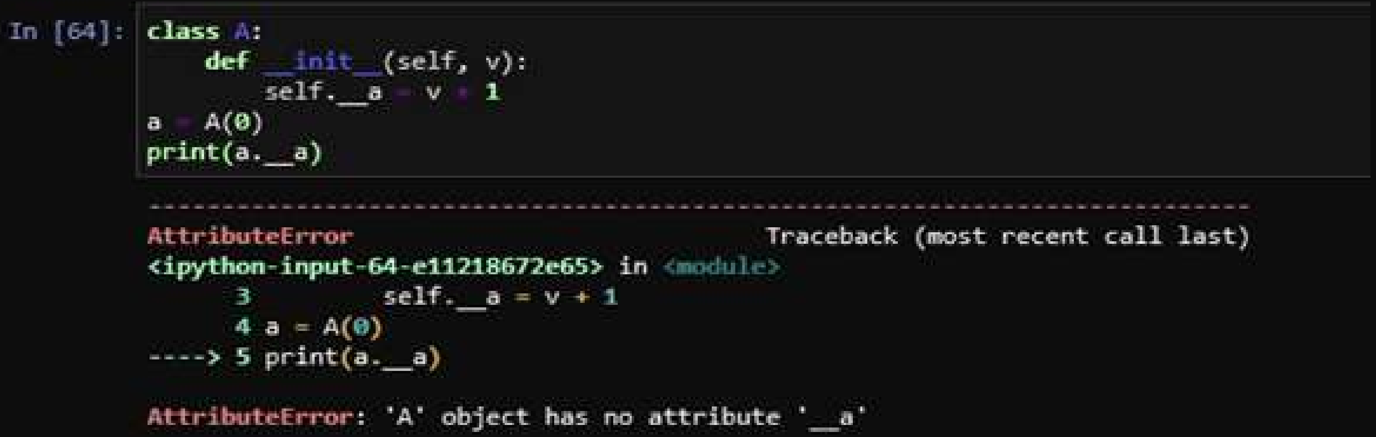
```
class A:
def __init__(self):
self.a = 1
class B:
def __init__(self):
# put selected line here
self.a = 2
```

- __init__()
- A.__init__()
- [✓ A.__init__(self)]
- A.__init__(1)

4. **What will be the effect of running the following code?**

```
class A:
def __init__(self,v):
self.__a = v + 1
a = A(0)
print(a.__a)
```

- it will print 0
- it will print 2
- it will print 1
- [✓ it will raise an AttributeError exception]



```
In [64]: class A:
             def __init__(self, v):
                 self.__a = v + 1
         a = A(0)
         print(a.__a)

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-64-e11218672e65> in <module>
      3         self.__a = v + 1
      4 a = A(0)
----> 5 print(a.__a)

AttributeError: 'A' object has no attribute '__a'
```

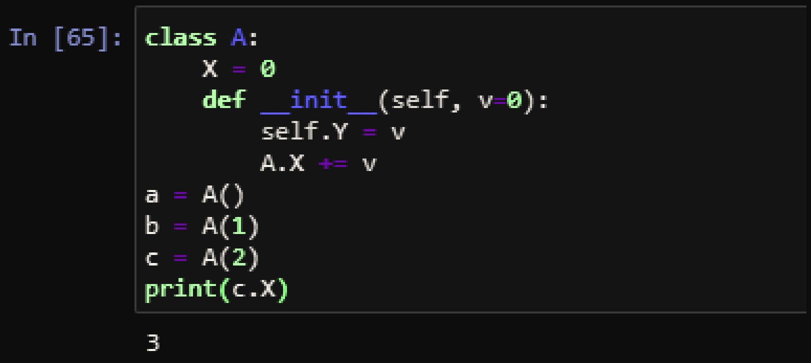
5. **What will be the output of the following code?**

```
class A:
def __init__(self,v = 1):
self.v = v
def set(self,v):
self.v = v
return v
a = A()
print(a.set(a.v + 1))
```

- 3
- 0
- 1
- [✓ 2]

6. **What will be the output of the following code?**

```
class A:
X = 0
def __init__(self,v = 0):
self.Y = v
A.X += v
a = A()
b = A(1)
c = A(2)
print(c.X)
```



```
In [65]: class A:
             X = 0
             def __init__(self, v=0):
                 self.Y = v
                 A.X += v
         a = A()
         b = A(1)
         c = A(2)
         print(c.X)

         3
```

- 0
- 2
- [✓ 3]
- 1

7. **What will be the output of the following code?**

class A:

A = 1

print(hasattr(A,’A’))

- 0
- False
- 1
- True

8. **What will be the result of executing the following code?**

class A:

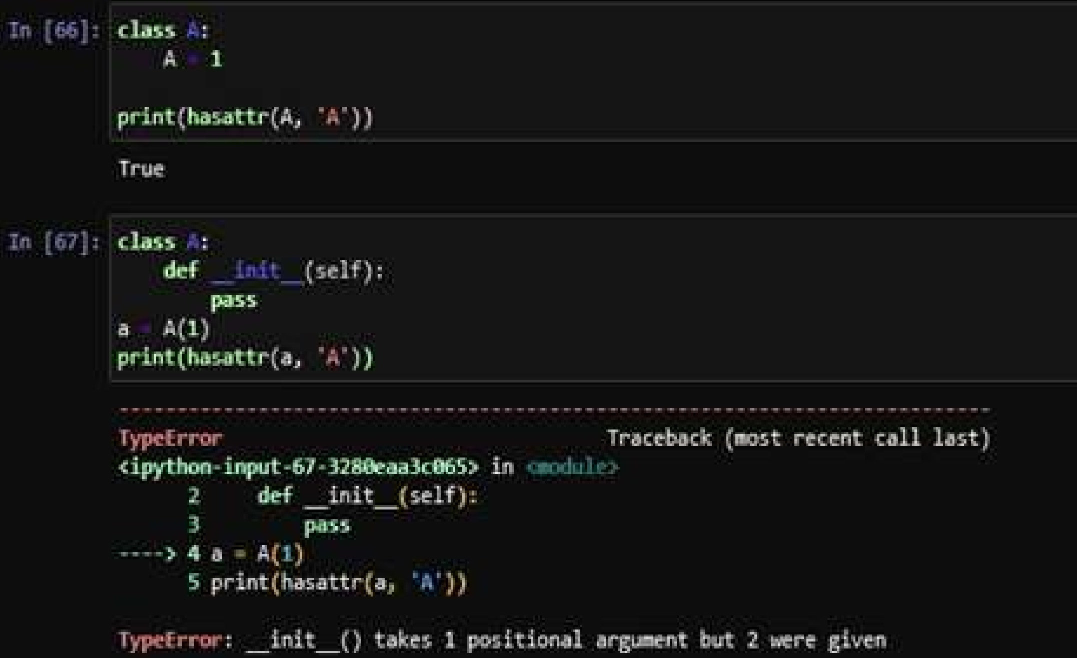
def __init__(self):

pass

a = A(1)

print(hasattr(a,’A’))

- it will print False
- it will print 1
- it will print True
- it will raise an exception



```
In [66]: class A:
             A = 1

         print(hasattr(A, 'A'))

         True

In [67]: class A:
             def __init__(self):
                 pass
         a = A(1)
         print(hasattr(a, 'A'))

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         <ipython-input-67-3280eaa3c065> in <module>
               2     def __init__(self):
               3         pass
         ----> 4 a = A(1)
               5 print(hasattr(a, 'A'))

         TypeError: __init__() takes 1 positional argument but 2 were given
```

9. **What will be the result of executing the following code?**

class A:

def __str__(self):

return ‘a’

class B(A):

def __str__(self):

return ‘b’

class C(B):

pass

o = C()

print(o)

- it will raise an exception
- it will print a
- it will print c
- [✓ it will print b]

10. **What will be the result of executing the following code?**

class A:

pass

class B(A):

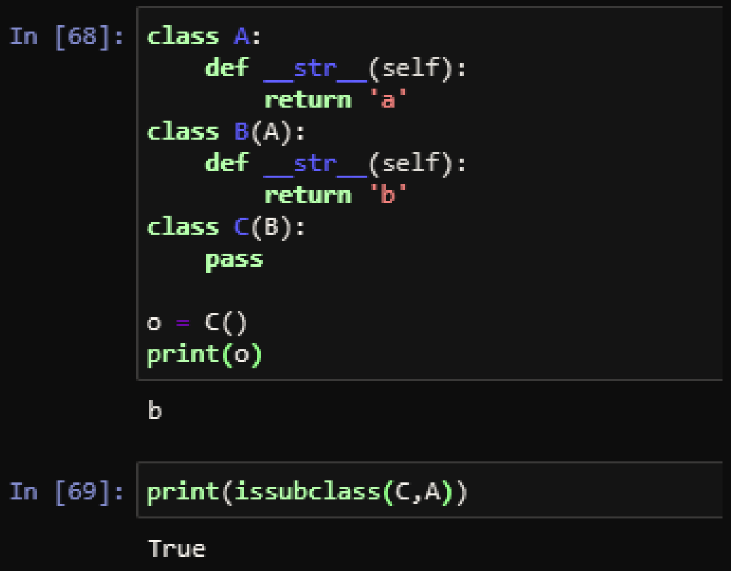
pass

class C(B):

pass

print(issubclass(C,A))

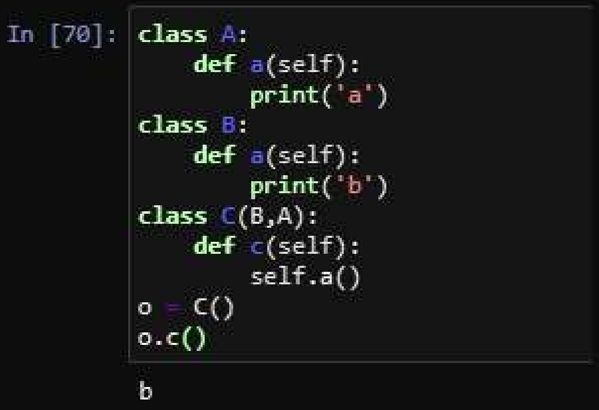
- it will raise an exception
- [✓ it will print True]
- it will print 1
- it will print False



```
In [68]: class A:
             def __str__(self):
                 return 'a'
         class B(A):
             def __str__(self):
                 return 'b'
         class C(B):
             pass

         o = C()
         print(o)

         b

In [69]: print(issubclass(C,A))

         True
```

11. **What will be the result of executing the following code?**

```
class A:
def a(self):
print('a')
class B:
def a(self):
print('b')
class C(B,A):
def c(self):
self.a()
o = C()
o.c()
```



```
In [70]: class A:
             def a(self):
                 print('a')
         class B:
             def a(self):
                 print('b')
         class C(B,A):
             def c(self):
                 self.a()
         o = C()
         o.c()

         b
```

- it will print c
- it will print a
- it will raise an exception
- [✓ it will print b]

12. **What will be the result of executing the following code?**

```
class A:
def __str__(self):
return 'a'
class B(A):
def __str__(self):
return 'b'
class C(B):
pass
o = C()
print(o)
```

- [✓ it will print b]
- it will raise an exception
- it will print a
- it will print c

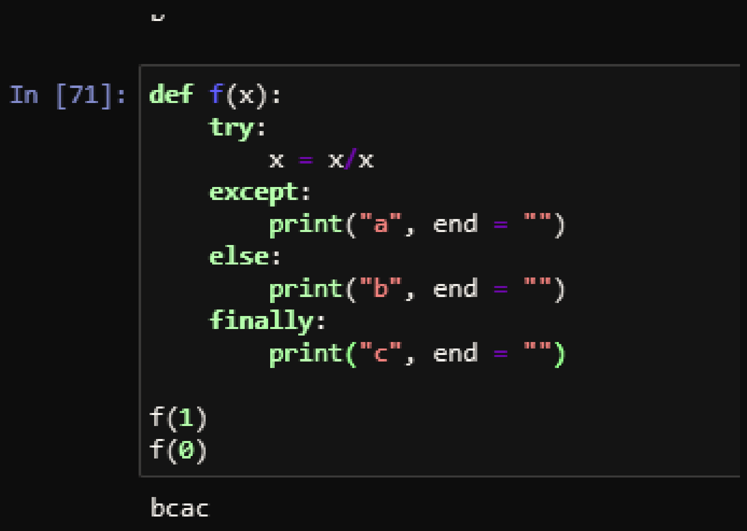
13. **What will be the result of executing the following code?**

```
class A:
v = 2
class B(A):
v = 1
class C(B):
pass
o = C()
print(o.v)
```

- it will print an empty line
- it will print 2
- it will raise an exception
- [✓ it will print 1]

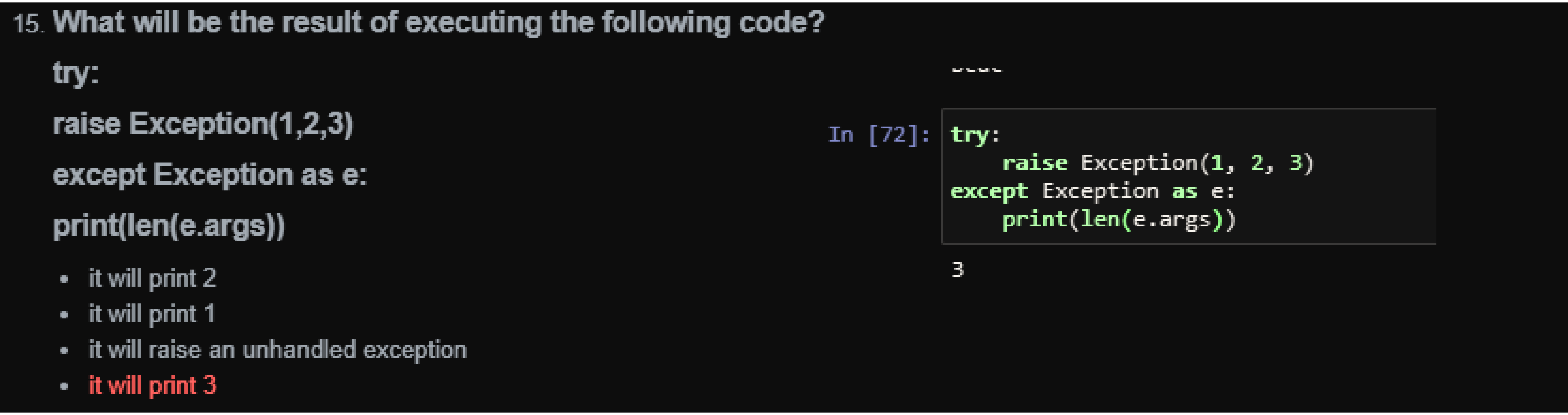
14. **What will be the result of executing the following code?**

```
def f(x):
try:
x = x / x
except:
print(“a”,end=”)
else:
print(“b”,end=”)
finally:
print(“c”,end=”)
f(1)
f(0)
```



```
In [71]: def f(x):
             try:
                 x = x/x
             except:
                 print("a", end = "")
             else:
                 print("b", end = "")
             finally:
                 print("c", end = "")

         f(1)
         f(0)

         bcac
```

- it will print bcbc
- [✓ it will print bcac]
- it will print acac
- it will raise an unhandled exception]



15. **What will be the result of executing the following code?**

try:

raise Exception(1,2,3)

except Exception as e:

print(len(e.args))

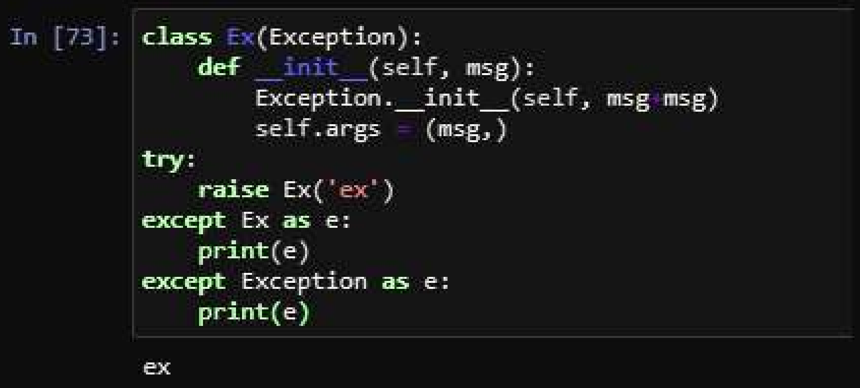
- it will print 2
- it will print 1
- it will raise an unhandled exception
- [✓ it will print 3]

```
In [72]: try:
             raise Exception(1, 2, 3)
         except Exception as e:
             print(len(e.args))
```

```
3
```

16. **What will be the result of executing the following code?**

```
class Ex(Exception):
def __init__(self,msg):
Exception.__init__(self,msg + msg)
self.args = (msg,)
try:
raise Ex('ex')
except Ex as e:
print(e)
except Exception as e:
print(e)
```

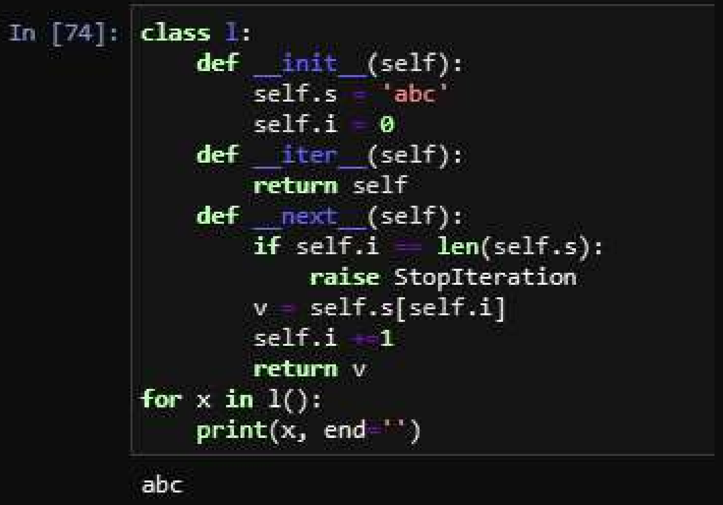


```
In [73]: class Ex(Exception):
             def __init__(self, msg):
                 Exception.__init__(self, msg+msg)
                 self.args = (msg,)
         try:
             raise Ex('ex')
         except Ex as e:
             print(e)
         except Exception as e:
             print(e)

ex
```

- it will raise an unhandled exception
- it will print an empty line
- it will print exex
- it will print ex

17. **What will be the result of executing the following code?**

```
class I:
def __init__(self):
self.s = 'abc'
self.i = 0
def __iter__(self):
return self
def __next__(self):
if self.i == len(self.s):
raise StopIteration
v = self.s[self.i]
self.i += 1
return v
for x in I():
print(x,end='')
```



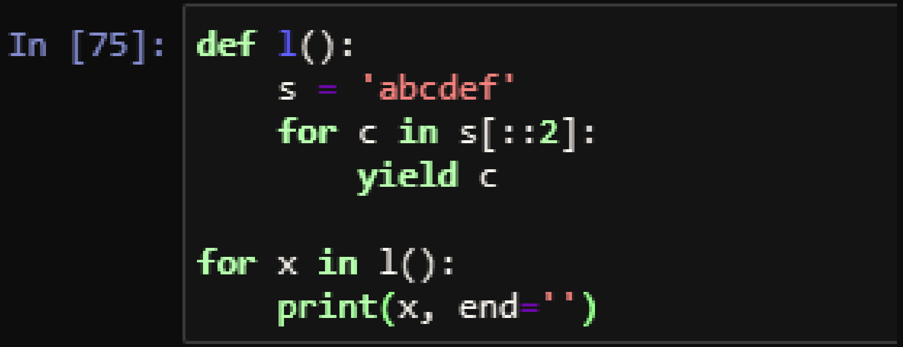
```
In [74]: class I:
             def __init__(self):
                 self.s = 'abc'
                 self.i = 0
             def __iter__(self):
                 return self
             def __next__(self):
                 if self.i == len(self.s):
                     raise StopIteration
                 v = self.s[self.i]
                 self.i +=1
                 return v
         for x in I():
             print(x, end='')

abc
```

- it will print cba
- it will print 210
- it will print 012
- it will print abc

18. **What will be the result of executing the following code?**

```
def l():
s = ‘abcdef’
for c in s[::2]:
yield c
for x in l():
print(x,end=”)
```

- it will print an empty line
- it will print bdf
- it will print abcdef
- [✓ it will print ace]



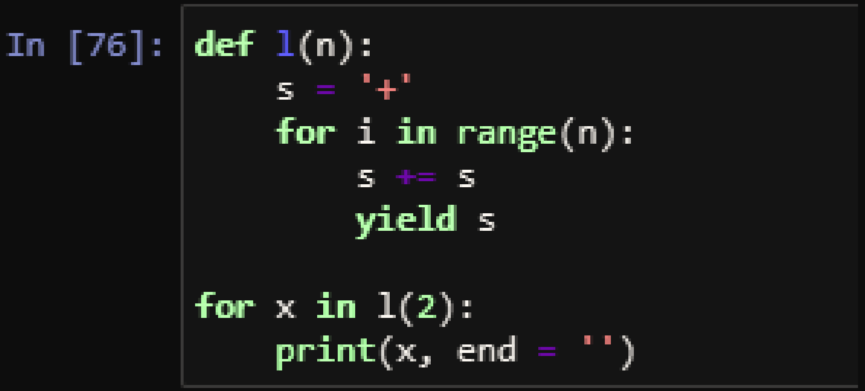
```
In [75]: def l():
             s = 'abcdef'
             for c in s[::2]:
                 yield c

         for x in l():
             print(x, end='')
```

```
ace
```

19. **What will be the result of executing the following code?**

```
def l(n):
s = ‘+’
for i in range(n):
s += s
yield s
for x in l(2):
print(x,end=”)
```

- it will print +
- it will print +++
- [✓ it will print ++++++]
- it will print ++



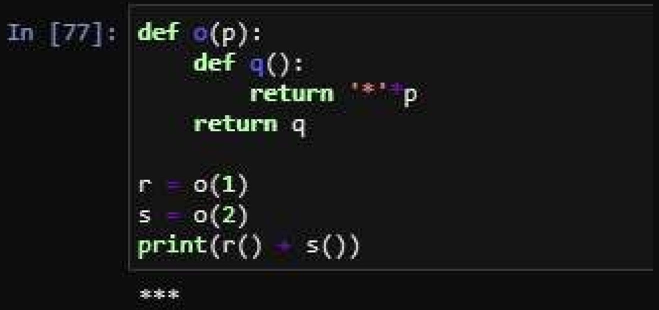
```
In [76]: def l(n):
             s = '+'
             for i in range(n):
                 s += s
                 yield s

         for x in l(2):
             print(x, end = '')
```

```
++++++
```

20. **What will be the result of executing the following code?**

```
def o(p):
def q():
return '*' * p
return q
r = o(1)
s = o(2)
print(r() + s())
```

- it will print ***
- it will print **
- it will print ****
- it will print *



```
In [77]: def o(p):
             def q():
                 return '*'*p
             return q

         r = o(1)
         s = o(2)
         print(r() + s())

         ***
```

21. **When a file is opened in read mode, it:**
- it must exist (an exception will be raised otherwise)
- it cannot exist (it has to be created every time)
- it will be deleted if it exists
- it doesn't have to exist (it will be created if absent)

22. **If you want to open a text file in append mode, you would use the following mode string:**
- t+a
- at
- a+t
- at+

23. **The sys.stdin stream is normally associated with a:**
- null device
- keyboard
- printer
- screen

24. **The strerror function comes from the OS module, and it's designed to:**
- raise a string exception
- translate an error description from one language to another
- translate an error description into an error number
- translate an error number into an error description

25. **If s is a stream opened in read mode, the following line**

q = s.read(1)

will:
- read 1 buffer from the stream
- read 1 kilobyte from the stream
- read 1 character from the stream
- read 1 line from the stream

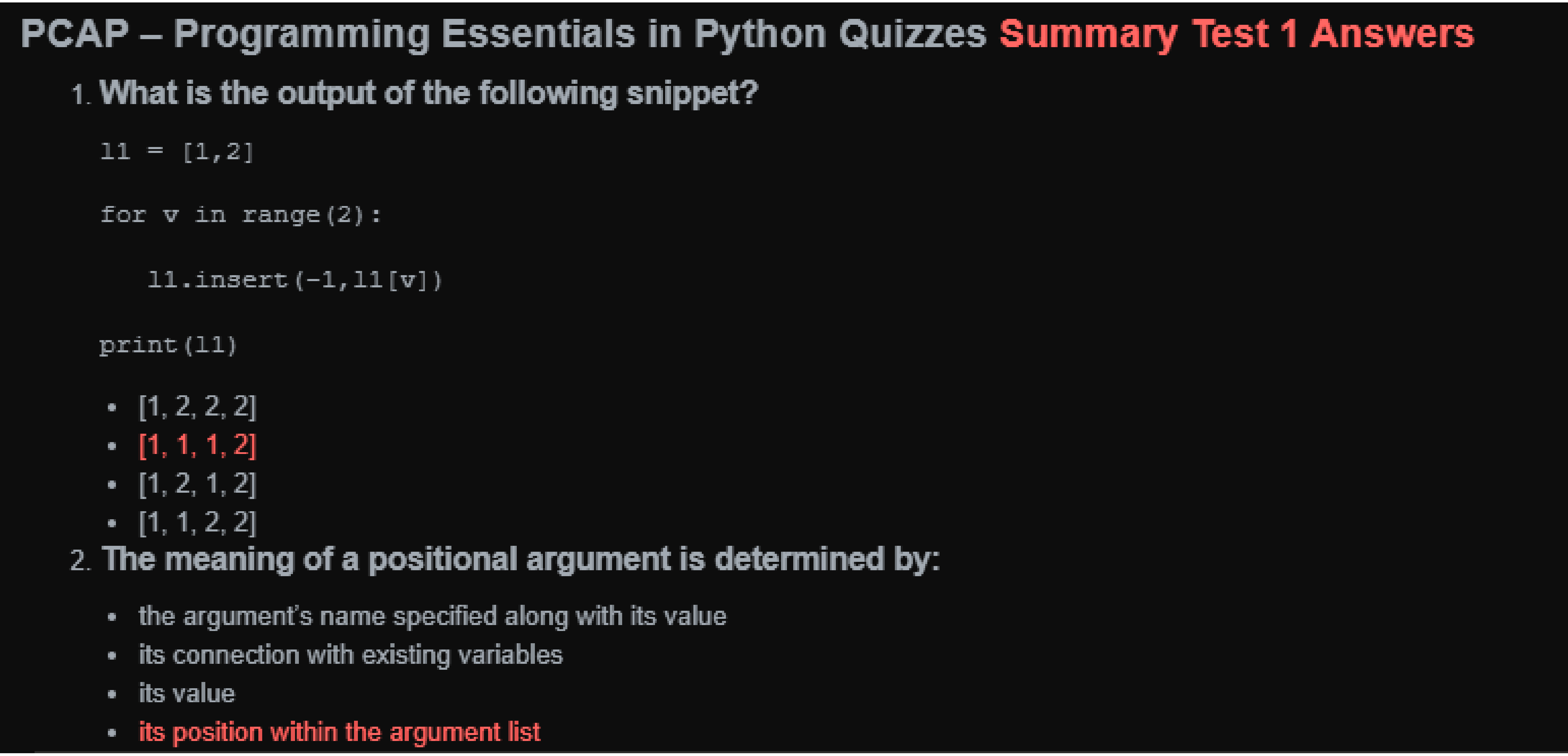
- read 1 line from the stream

26. **How does the readline() method react when the end-of-file occurs?**
    - it returns eof
    - it returns -1
    - it returns an empty string
    - it raises an exception
27. **The readlines() method returns a:**
    - list
    - dictionary
    - tuple
    - string
28. **Assuming that the open() invocation has gone successfully, the following snippet will:**

    for x in open('file','rt'):

    print(x)

    - read the file line by line
    - read the file character by character
    - cause an exception
    - read the whole file at once
29. **The byte array class can create objects which are designed to:**
    - build arrays 1 byte in size
    - convert tuples into lists
    - convert lists into tuples
    - store amorphic data organized in bytes
30. **If you want to fill a byte array with data read in from a stream, you use the:**
    - read() method
    - readinto() method
    - readfrom() method
    - readbytes() method



PCAP – Programming Essentials in Python Quizzes Summary Test 1 Answers

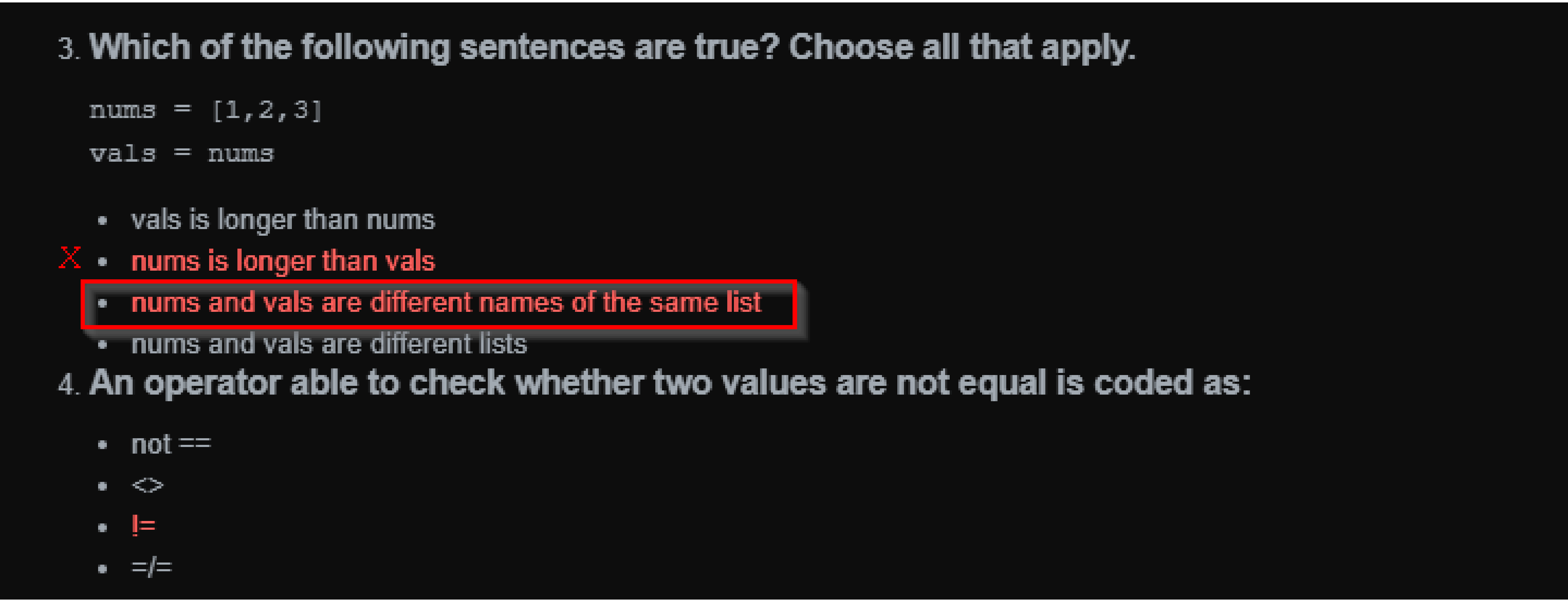
1. **What is the output of the following snippet?**

```
l1 = [1,2]

for v in range(2):

   l1.insert(-1,l1[v])

print(l1)
```

   - [1, 2, 2, 2]
   - [✓ [1, 1, 1, 2]]
   - [1, 2, 1, 2]
   - [1, 1, 2, 2]

2. **The meaning of a positional argument is determined by:**

   - the argument's name specified along with its value
   - its connection with existing variables
   - its value
   - [✓ its position within the argument list]



3. **Which of the following sentences are true? Choose all that apply.**

```
nums = [1,2,3]
vals = nums
```

- vals is longer than nums
- X nums is longer than vals
- [✓ nums and vals are different names of the same list]
- nums and vals are different lists

4. **An operator able to check whether two values are not equal is coded as:**

- not ==
- <>
- !=
- =/=

5. **The following snippet:**

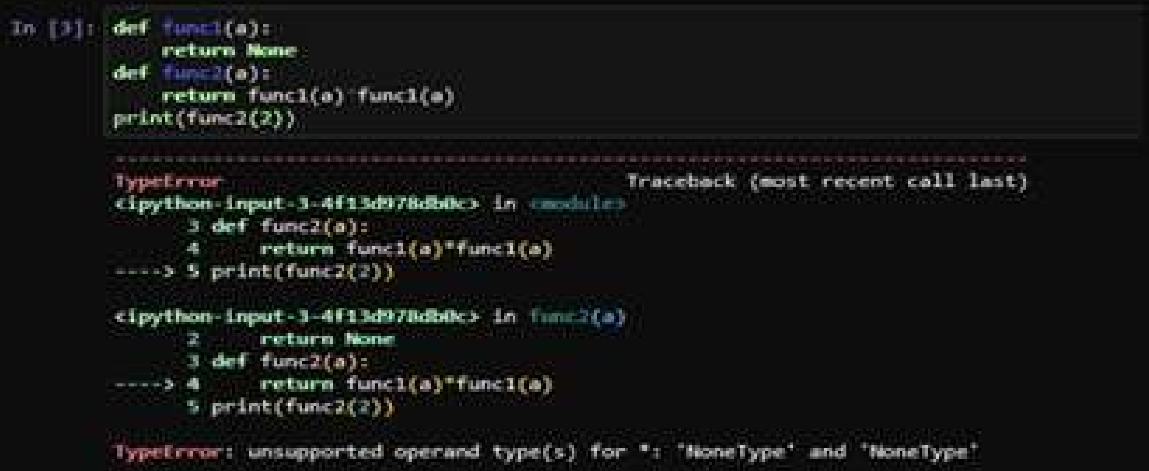
```
def func1(a):
    return None
def func2(a):
    return func1(a)*func1(a)
print(func2(2))
```

- will output 2
- will cause a runtime error
- will output 4
- will output 16

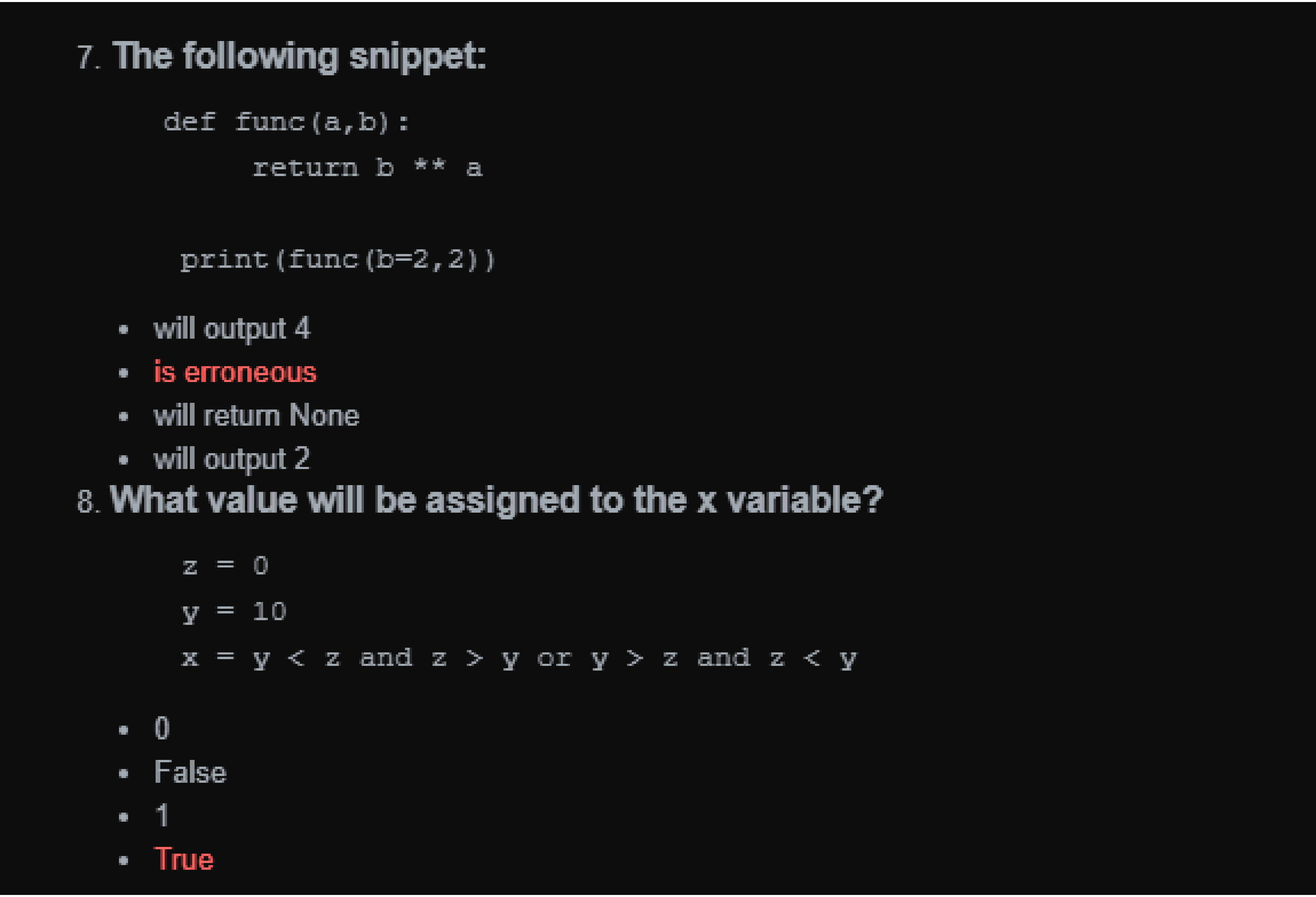
6. **The result of the following division:**

```
1 // 2
```

- cannot be predicted
- is equal to 0.5
- is equal to 0.0
- is equal to 0



```
In [3]: def func1(a):
            return None
        def func2(a):
            return func1(a)*func1(a)
        print(func2(2))

        ---------------------------------------------------------------------------
        TypeError                                 Traceback (most recent call last)
        <ipython-input-3-4f13d978db0c> in <module>
              3 def func2(a):
              4     return func1(a)*func1(a)
        ----> 5 print(func2(2))

        <ipython-input-3-4f13d978db0c> in func2(a)
              2     return None
              3 def func2(a):
        ----> 4     return func1(a)*func1(a)
              5 print(func2(2))

        TypeError: unsupported operand type(s) for *: 'NoneType' and 'NoneType'
```



7. **The following snippet:**

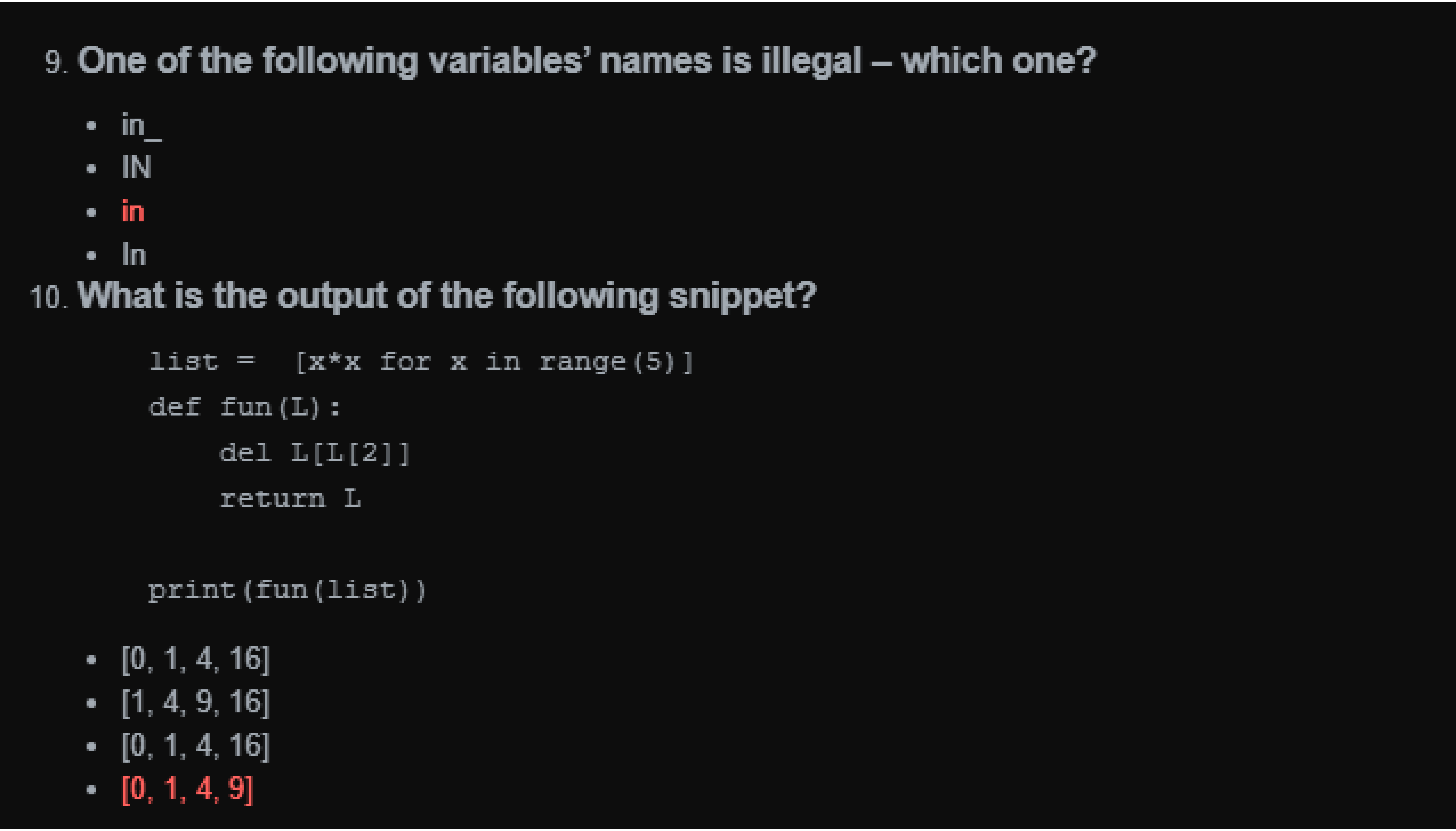
```
def func(a,b):
     return b ** a


 print(func(b=2,2))
```

- will output 4
- [✓ is erroneous]
- will return None
- will output 2

8. **What value will be assigned to the x variable?**

```
z = 0
y = 10
x = y < z and z > y or y > z and z < y
```

- 0
- False
- 1
- [✓ True]



9. **One of the following variables' names is illegal – which one?**

- in_
- IN
- [✓ in]
- In

10. **What is the output of the following snippet?**

```
list =  [x*x for x in range(5)]
def fun(L):
    del L[L[2]]
    return L


print(fun(list))
```

- [0, 1, 4, 16]
- [1, 4, 9, 16]
- [0, 1, 4, 16]
- [✓ [0, 1, 4, 9]]

11. **What is the output of the following piece of code?**

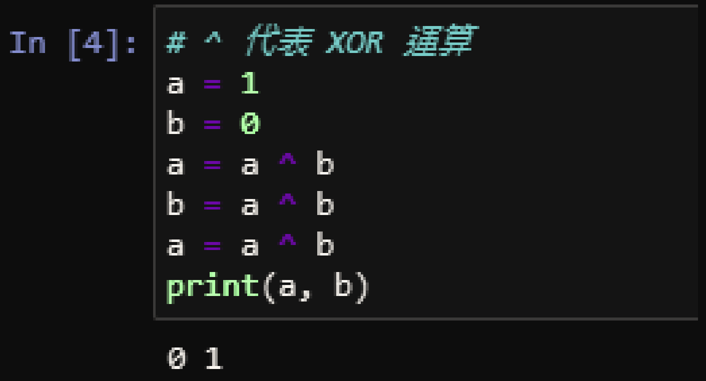
```
x=1
y=2
x, y, z = x, x, y
z, y, z = x, y, z
print(x,y,z)
```

- 1 1 2
- 1 2 2
- 2 1 2
- 1 2 1

12. **What will the output of the following snippet?**

```
a = 1
b = 0
a = a ^ b
b = a ^ b
a = a ^ b
print(a,b)
```

- 0 1
- 1 0
- 0 0
- 1 1



```
In [4]: # ^ 代表 XOR 運算
        a = 1
        b = 0
        a = a ^ b
        b = a ^ b
        a = a ^ b
        print(a, b)

        0 1
```

13. **What is the output of the following snippet?**

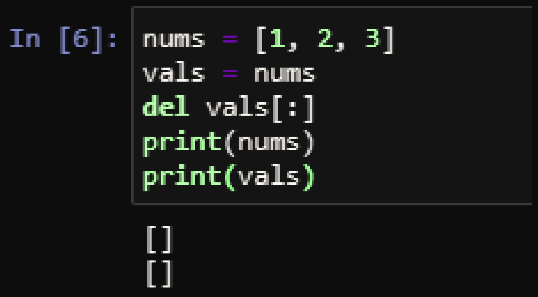
```
def fun(x):
    if x % 2 == 0:
        return 1
    else:
        return 2


print(fun(fun(2)))
```

- None
- 1
- the code will cause a runtime error
- [✓ 2]

14. **Take a look at the snippet and choose the true statement:**

```
nums = [1,2,3]
vals = nums
del vals[:]
```

- [✓ nums and vals are different names of the same list]
- vals is longer than nums
- the snippet will cause a runtime error
- nums and vals are different lists

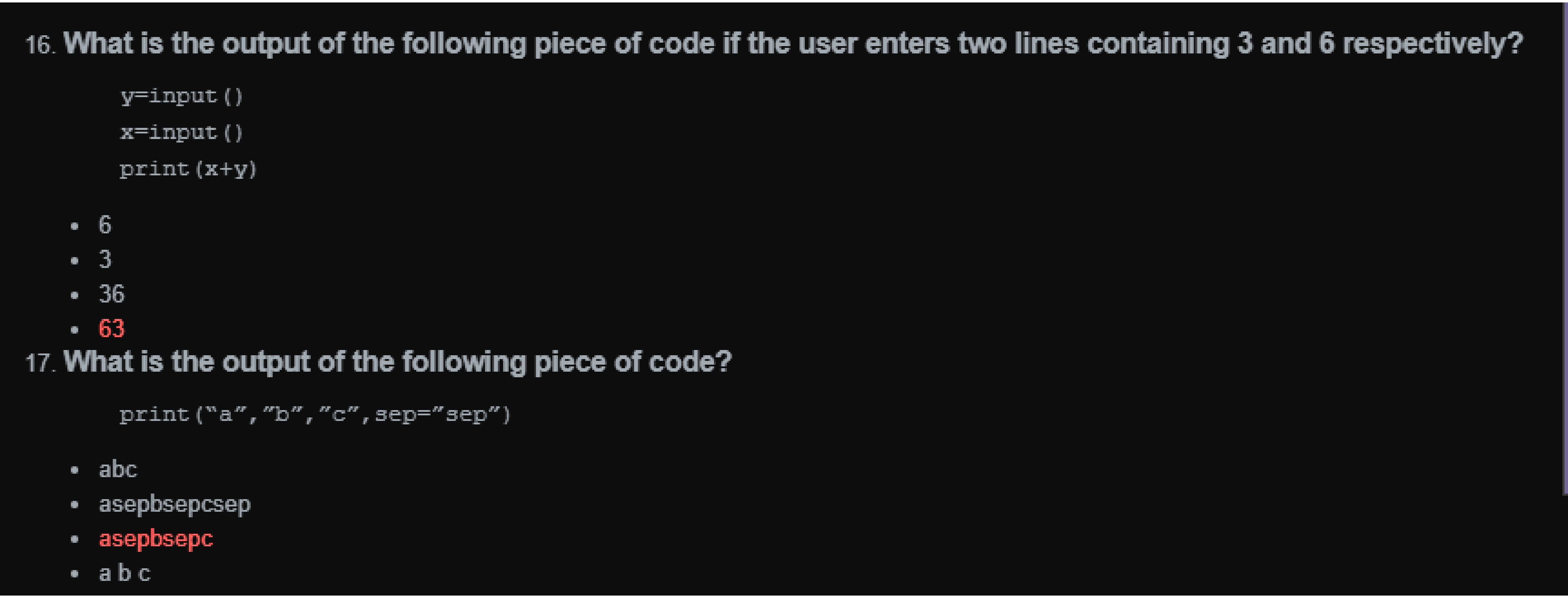


```
In [6]: nums = [1, 2, 3]
        vals = nums
        del vals[:]
        print(nums)
        print(vals)

        []
        []
```

15. **What is the output of the following piece of code if the user enters two lines containing 3 and 2 respectively?**

```
x=int(input())
y=int(input())
x = x % y
x = x % y
y = y % x
print(y)
```

- 2
- 1
- [✓ 0]
- 3



16. **What is the output of the following piece of code if the user enters two lines containing 3 and 6 respectively?**

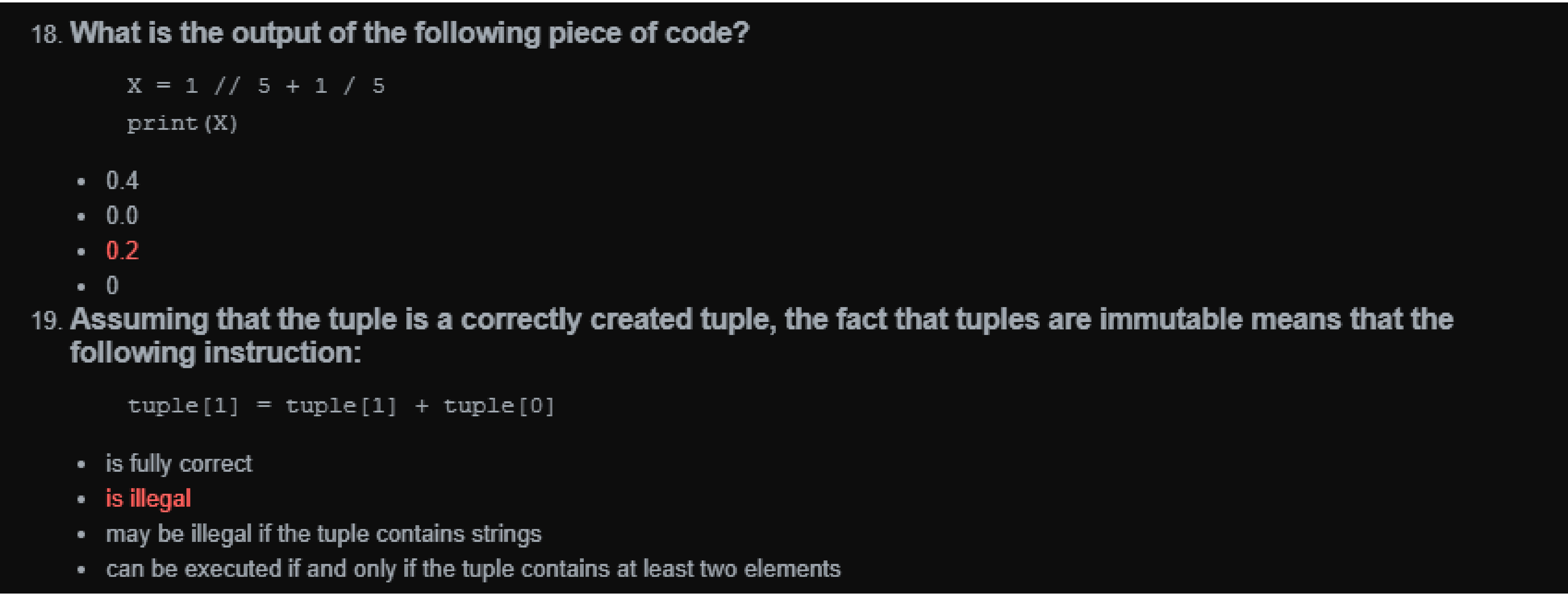
```
y=input()
x=input()
print(x+y)
```

- 6
- 3
- 36
- [✓ 63]

17. **What is the output of the following piece of code?**

```
print(“a”,”b”,”c”,sep=”sep”)
```

- abc
- asepbsepcsep
- [✓ asepbsepc]
- a b c



18. **What is the output of the following piece of code?**

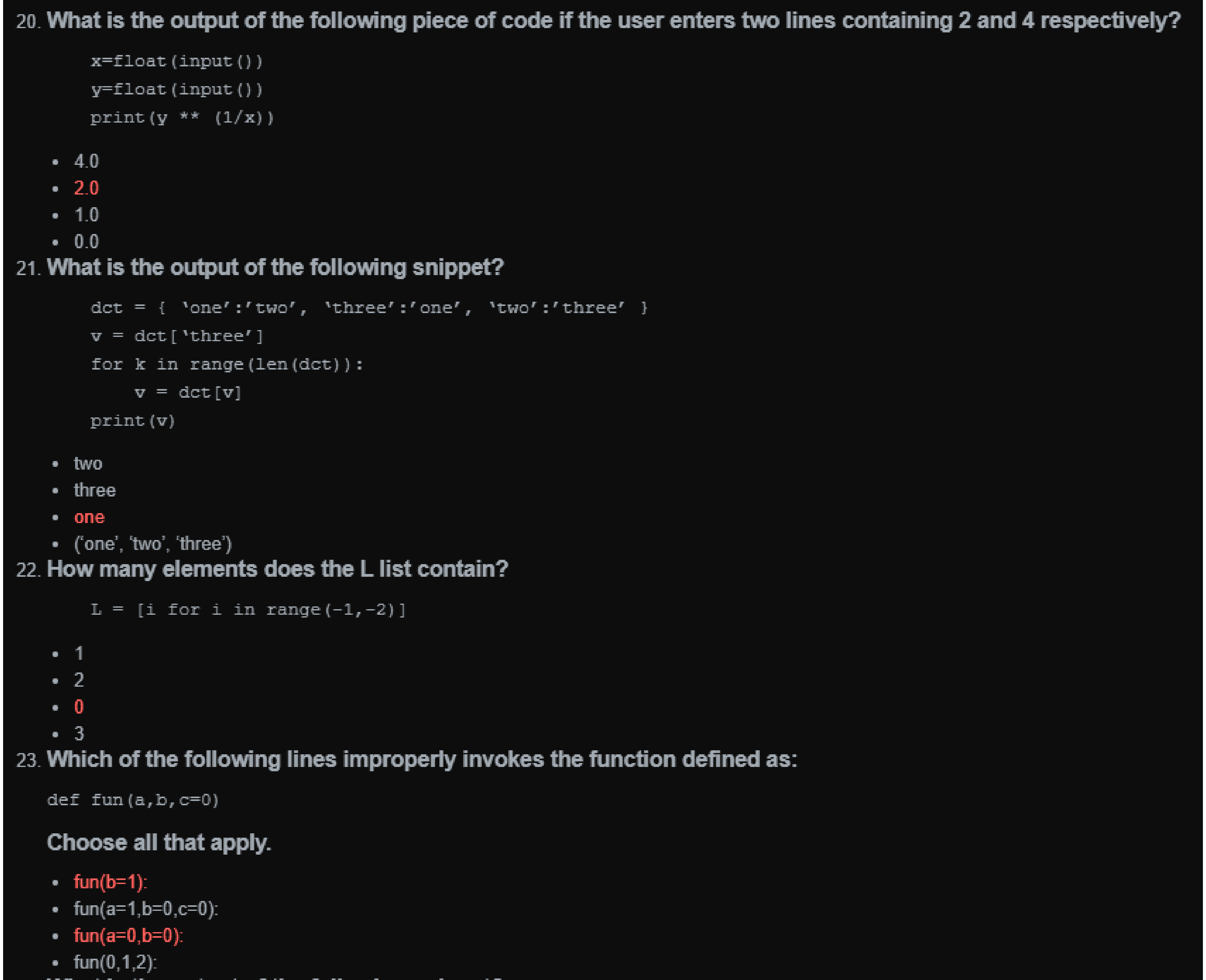
```
X = 1 // 5 + 1 / 5
print(X)
```

- 0.4
- 0.0
- [✓ 0.2]
- 0

19. **Assuming that the tuple is a correctly created tuple, the fact that tuples are immutable means that the following instruction:**

```
tuple[1] = tuple[1] + tuple[0]
```

- is fully correct
- [✓ is illegal]
- may be illegal if the tuple contains strings
- can be executed if and only if the tuple contains at least two elements



20. **What is the output of the following piece of code if the user enters two lines containing 2 and 4 respectively?**

```
x=float(input())
y=float(input())
print(y ** (1/x))
```

- 4.0
- [✓ 2.0]
- 1.0
- 0.0

21. **What is the output of the following snippet?**

```
dct = { ‘one’:’two’, ‘three’:’one’, ‘two’:’three’ }
v = dct[‘three’]
for k in range(len(dct)):
    v = dct[v]
print(v)
```

- two
- three
- [✓ one]
- (‘one’, ‘two’, ‘three’)

22. **How many elements does the L list contain?**

```
L = [i for i in range(-1,-2)]
```

- 1
- 2
- [✓ 0]
- 3

23. **Which of the following lines improperly invokes the function defined as:**

```
def fun(a,b,c=0)
```

Choose all that apply.

- [✓ fun(b=1):]
- fun(a=1,b=0,c=0):
- [✓ fun(a=0,b=0):]
- fun(0,1,2):

24. **What is the output of the following snippet?**

```
def fun(x,y):
    if x == y:
        return x
    else:
        return fun(x,y-1)


print(fun(0,3))
```

- [✓ 0]
- 1
- the snippet will cause a runtime error
- 2

25. **How many stars will the following snippet send to the console?**

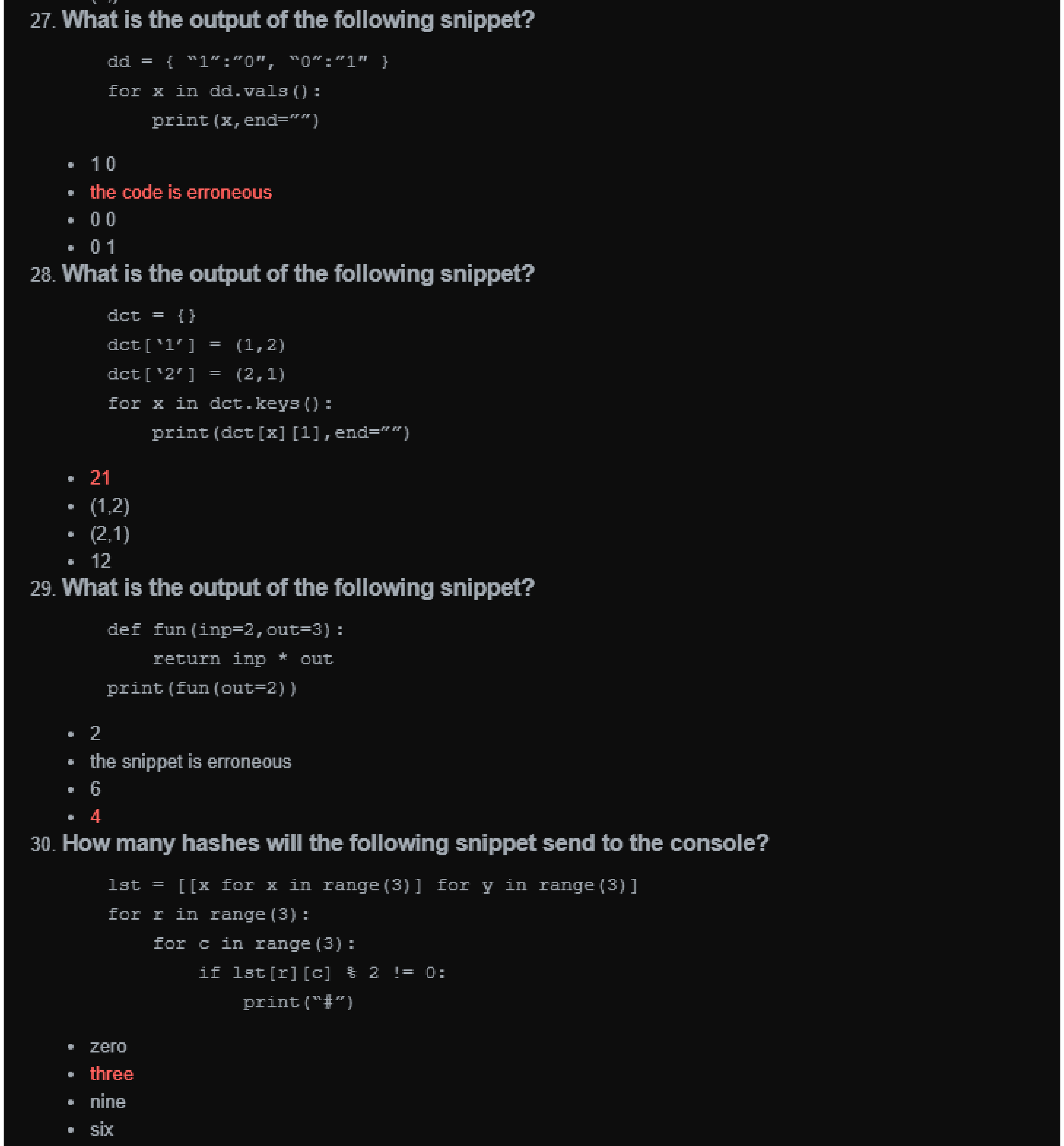
```
i = 0
while i < i + 2 :
    i += 1
    print("*")
else:
    print("*")
```

- [✓ the snippet will enter an infinite loop]
- zero
- one
- two

26. **What is the output of the following snippet?**

```
tup = (1, 2, 4, 8)
tup = tup[-2:-1]
tup = tup[-1]
print(tup)
```

- [✓ 4]
- (4)
- 44
- (4,)



27. **What is the output of the following snippet?**

```
dd = { “1”:”0”, “0”:”1” }
for x in dd.vals():
    print(x,end=””)
```

- 1 0
- [✓ the code is erroneous]
- 0 0
- 0 1

28. **What is the output of the following snippet?**

```
dct = {}
dct[‘1’] = (1,2)
dct[‘2’] = (2,1)
for x in dct.keys():
    print(dct[x][1],end=””)
```

- [✓ 21]
- (1,2)
- (2,1)
- 12

29. **What is the output of the following snippet?**

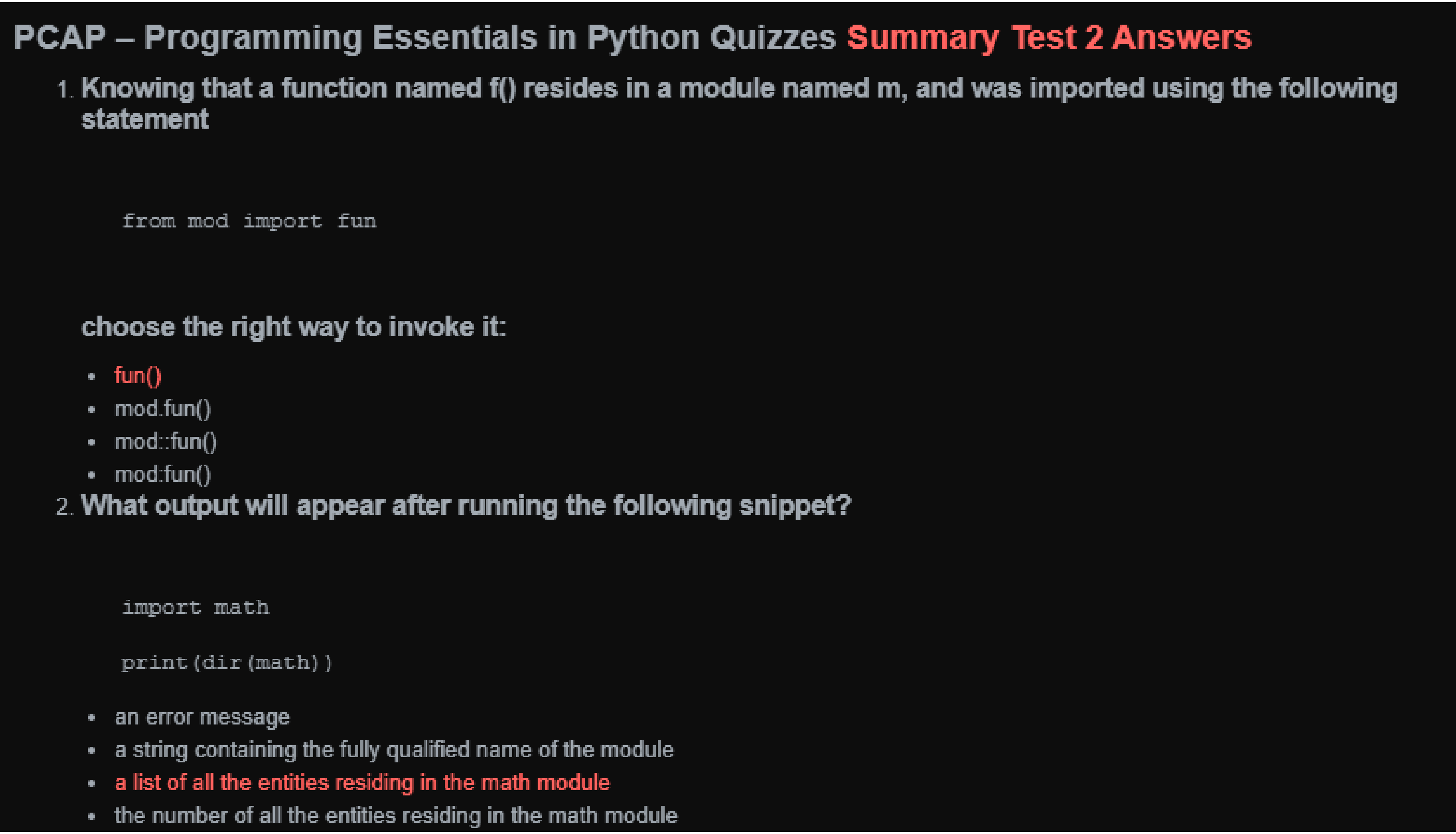
```
def fun(inp=2,out=3):
    return inp * out
print(fun(out=2))
```

- 2
- the snippet is erroneous
- 6
- [✓ 4]

30. **How many hashes will the following snippet send to the console?**

```
lst = [[x for x in range(3)] for y in range(3)]
for r in range(3):
    for c in range(3):
        if lst[r][c] % 2 != 0:
            print(“#”)
```

- zero
- [✓ three]
- nine
- six



PCAP – Programming Essentials in Python Quizzes Summary Test 2 Answers

1. **Knowing that a function named f() resides in a module named m, and was imported using the following statement**

```
from mod import fun
```

choose the right way to invoke it:

- [✓ fun()]
- mod.fun()
- mod::fun()
- mod:fun()

2. **What output will appear after running the following snippet?**

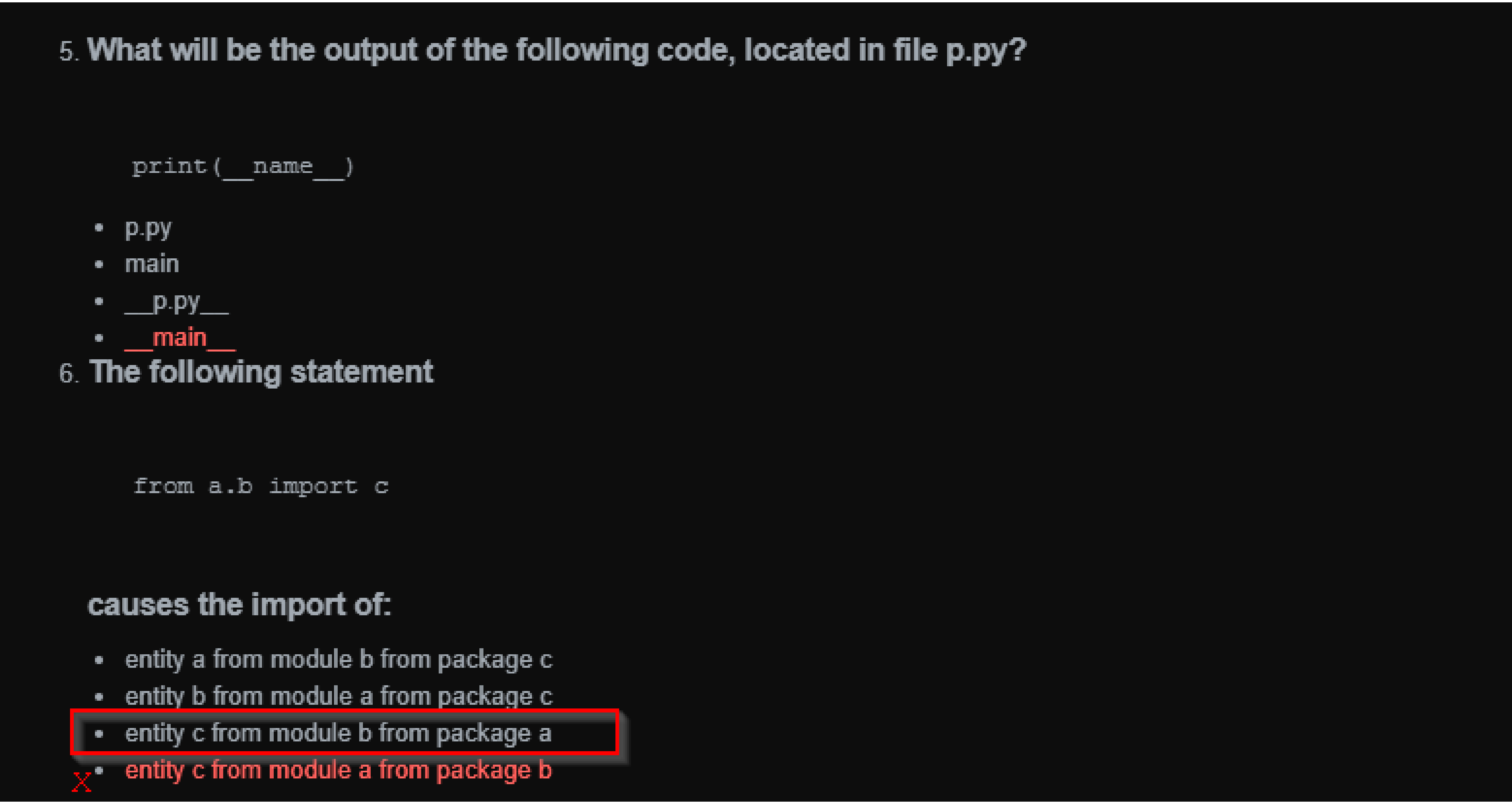
```
import math

print(dir(math))
```

- an error message
- a string containing the fully qualified name of the module
- [✓ a list of all the entities residing in the math module]
- the number of all the entities residing in the math module

3. **The compiled Python bytecode is stored in files having names ending with:**
   - py
   - pyb
   - pc
   - [✓ pyc]
4. **Assuming that all three files, a.py, b.py, and c.py reside in the same folder, what will be the output produced by running the c.py file?**

```
# file a.py

print(“a”,end=”)



#file b.py

import a

print(“b”,end=”)



#file c.py

print(“c”,end=”)

import a

import b
```

   - cba
   - abc
   - bac
   - [✓ cab]



5. **What will be the output of the following code, located in file p.py?**

```
print(__name__)
```

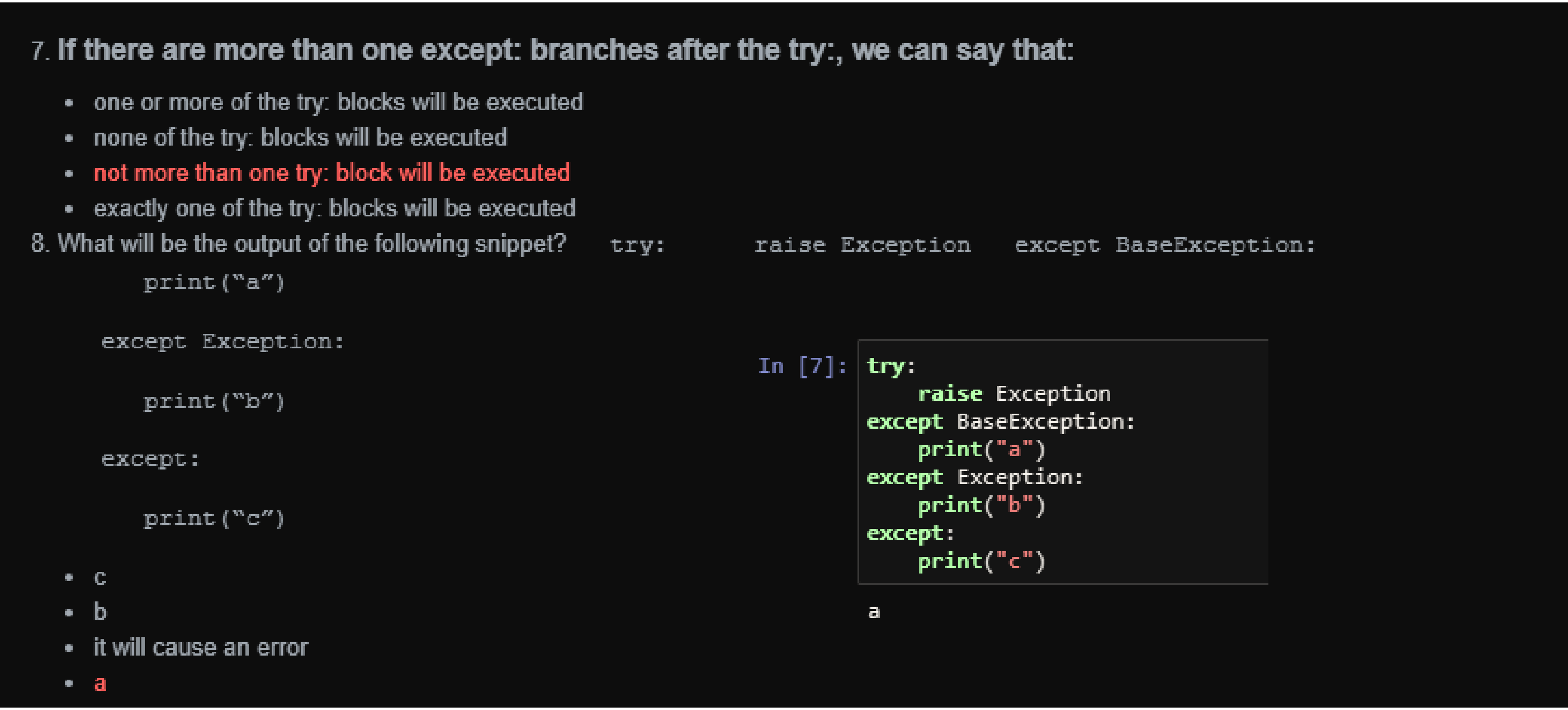
- p.py
- main
- __p.py__
- __main__

6. **The following statement**

```
from a.b import c
```

causes the import of:

- entity a from module b from package c
- entity b from module a from package c
- [✓ entity c from module b from package a]
- X entity c from module a from package b



7. **If there are more than one except: branches after the try:, we can say that:**

- one or more of the try: blocks will be executed
- none of the try: blocks will be executed
- not more than one try: block will be executed
- exactly one of the try: blocks will be executed

8. What will be the output of the following snippet? `try:        raise Exception    except BaseException:`

```
        print("a")

    except Exception:

        print("b")

    except:

        print("c")
```

- c
- b
- it will cause an error
- a

```
In [7]: try:
            raise Exception
        except BaseException:
            print("a")
        except Exception:
            print("b")
        except:
            print("c")
```

```
a
```

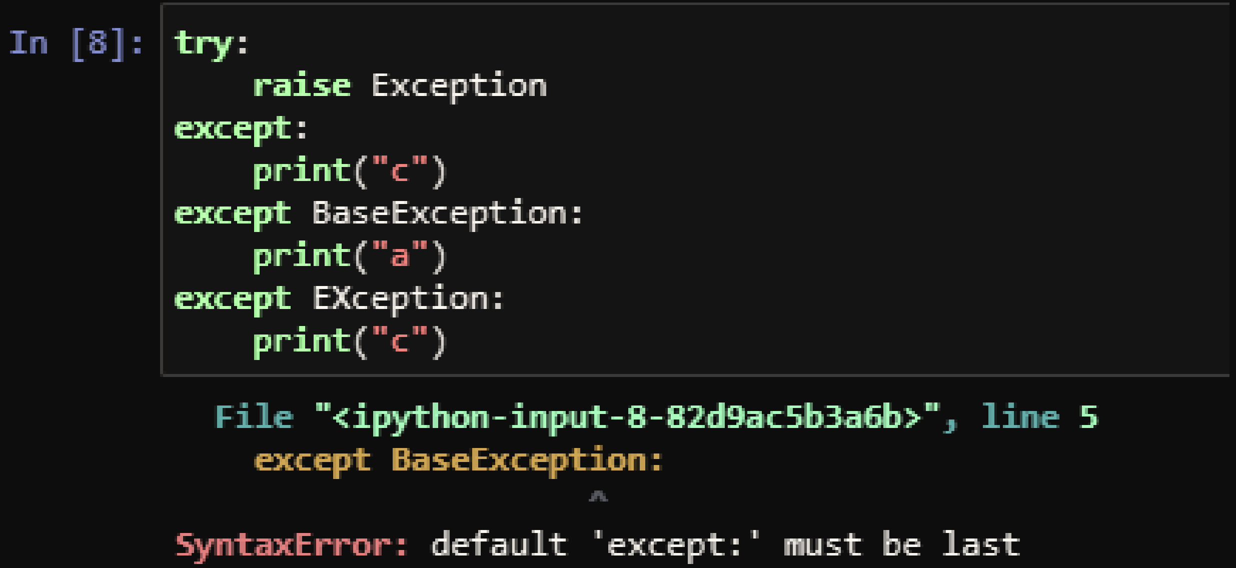
9. **The following line of code:**

```
for line in open(‘text.txt’,’rt’):
```

- [✓ is valid as open returns an iterable object]
- is invalid as open returns a non-iterable object
- is invalid as open returns nothing
- may be valid if line is a list

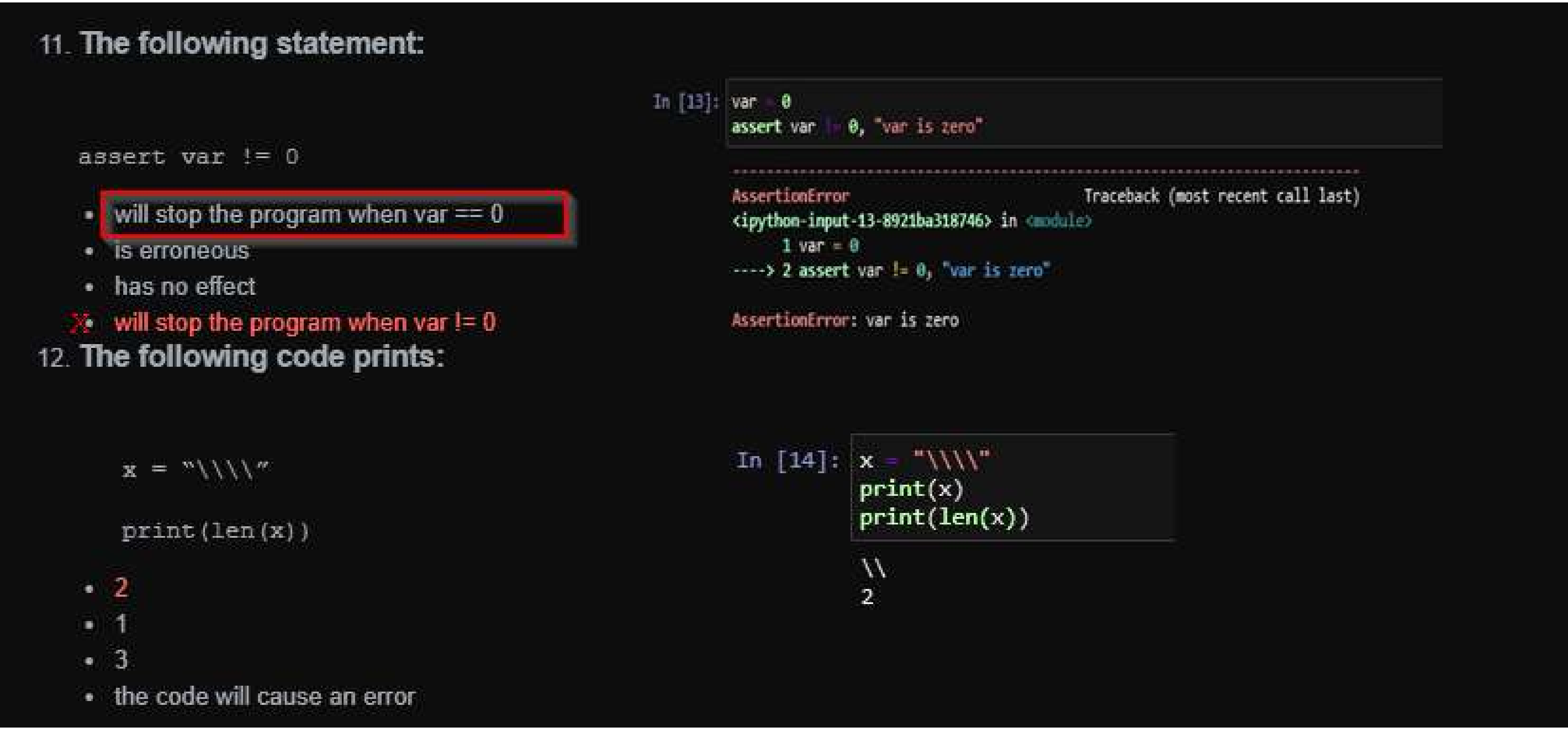
10. **What will be the output of the following snippet?**

```
try:

   raise Exception

except:

   print(“c”)

except BaseException:

   print(“a”)

except Exception:

   print(“b”)
```



```
In [8]: try:
            raise Exception
        except:
            print("c")
        except BaseException:
            print("a")
        except EXception:
            print("c")

          File "<ipython-input-8-82d9ac5b3a6b>", line 5
            except BaseException:
                                 ^
        SyntaxError: default 'except:' must be last
```

- a
- c
- b
- [✓ it will cause an error]



11. **The following statement:**

```
assert var != 0
```

- [✓ will stop the program when var == 0]
- is erroneous
- has no effect
- ✗ will stop the program when var != 0

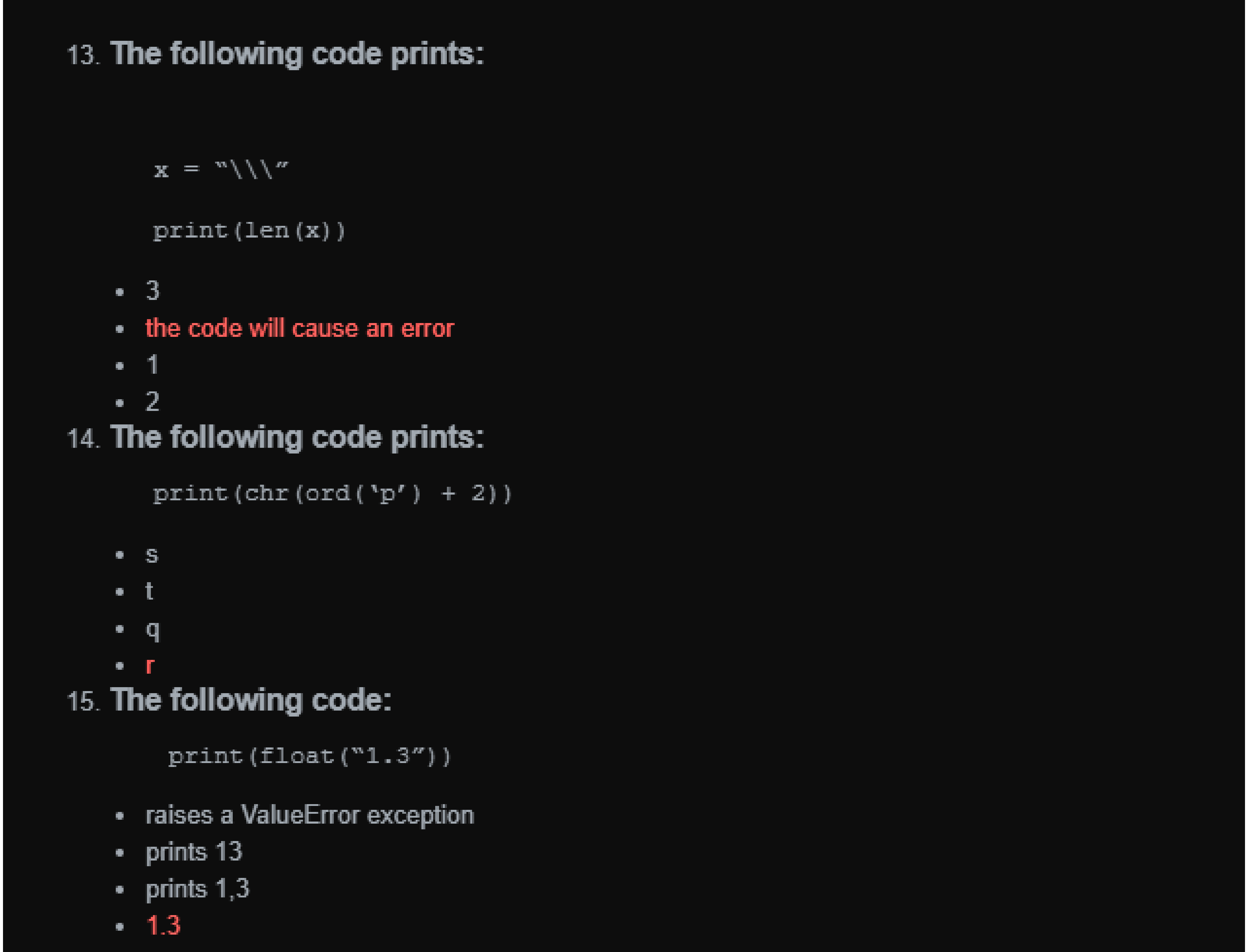
```
In [13]: var = 0
         assert var != 0, "var is zero"

---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-13-8921ba318746> in <module>
      1 var = 0
----> 2 assert var != 0, "var is zero"

AssertionError: var is zero
```

12. **The following code prints:**

```
x = “\\\\”

print(len(x))
```

- [✓ 2]
- 1
- 3
- the code will cause an error

```
In [14]: x = "\\\\"
         print(x)
         print(len(x))

\\
2
```



13. **The following code prints:**

```
x = “\\\”

print(len(x))
```

- 3
- [✓ the code will cause an error]
- 1
- 2

14. **The following code prints:**

```
print(chr(ord(‘p’) + 2))
```

- s
- t
- q
- [✓ r]

15. **The following code:**

```
print(float(“1.3”))
```

- raises a ValueError exception
- prints 13
- prints 1,3
- [✓ 1.3]

16. **If the class's constructor is declared as below, which one of the assignments is invalid?**

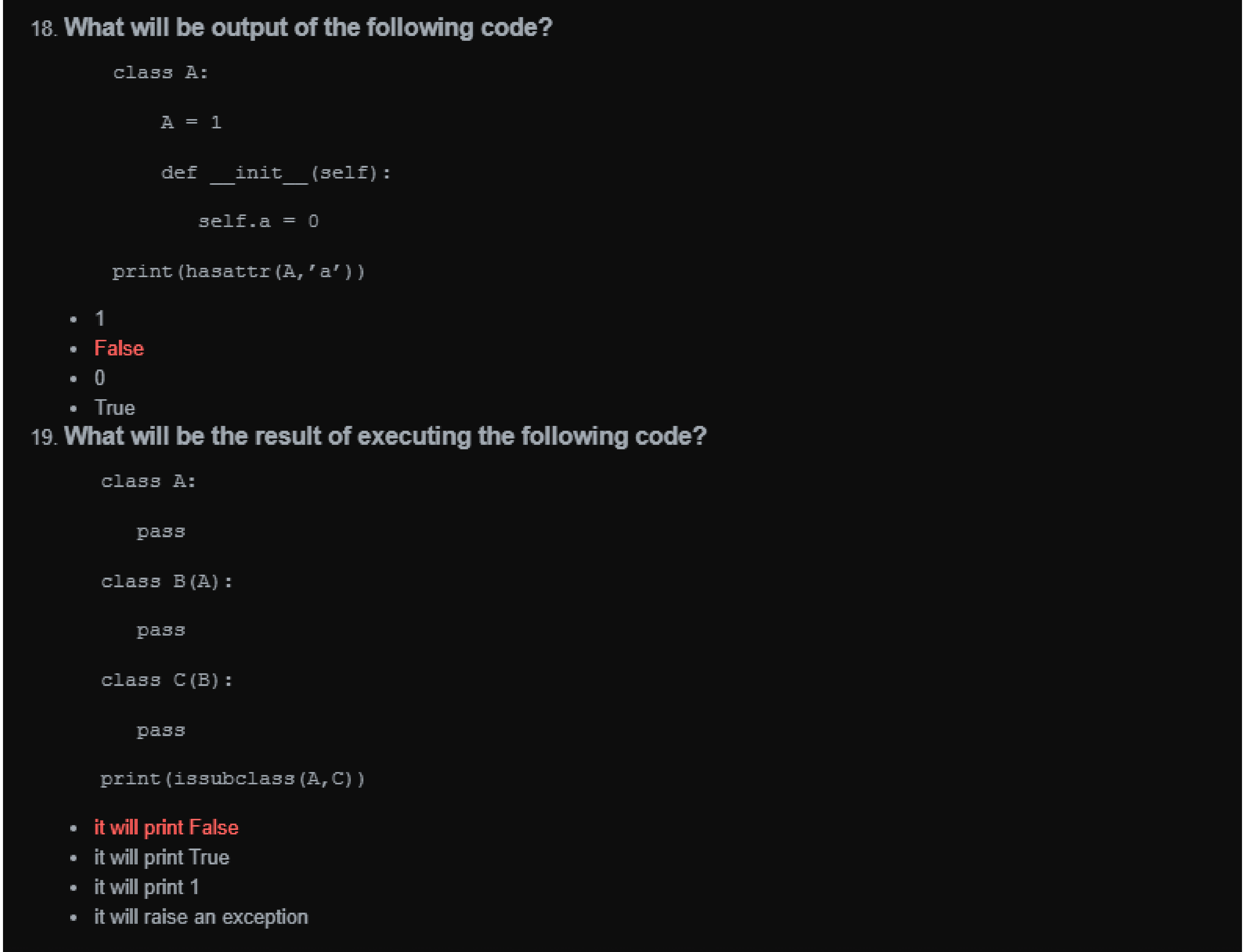
```
class Class:

   def __init__(self,val=0):

       pass
```

- object = Class(None)
- object = Class(1)
- object = Class()
- [✓ object = Class(1,2)]

17. **What will be output of the following code?**

```
class A:

   def __init__(self,v = 2):

      self.v = v

   def set(self,v = 1):

      self.v += v

       return self.v

a = A()

b = a

b.set()

print(a.v)
```

- 0
- [✓ 3]
- 1
- 2



18. **What will be output of the following code?**

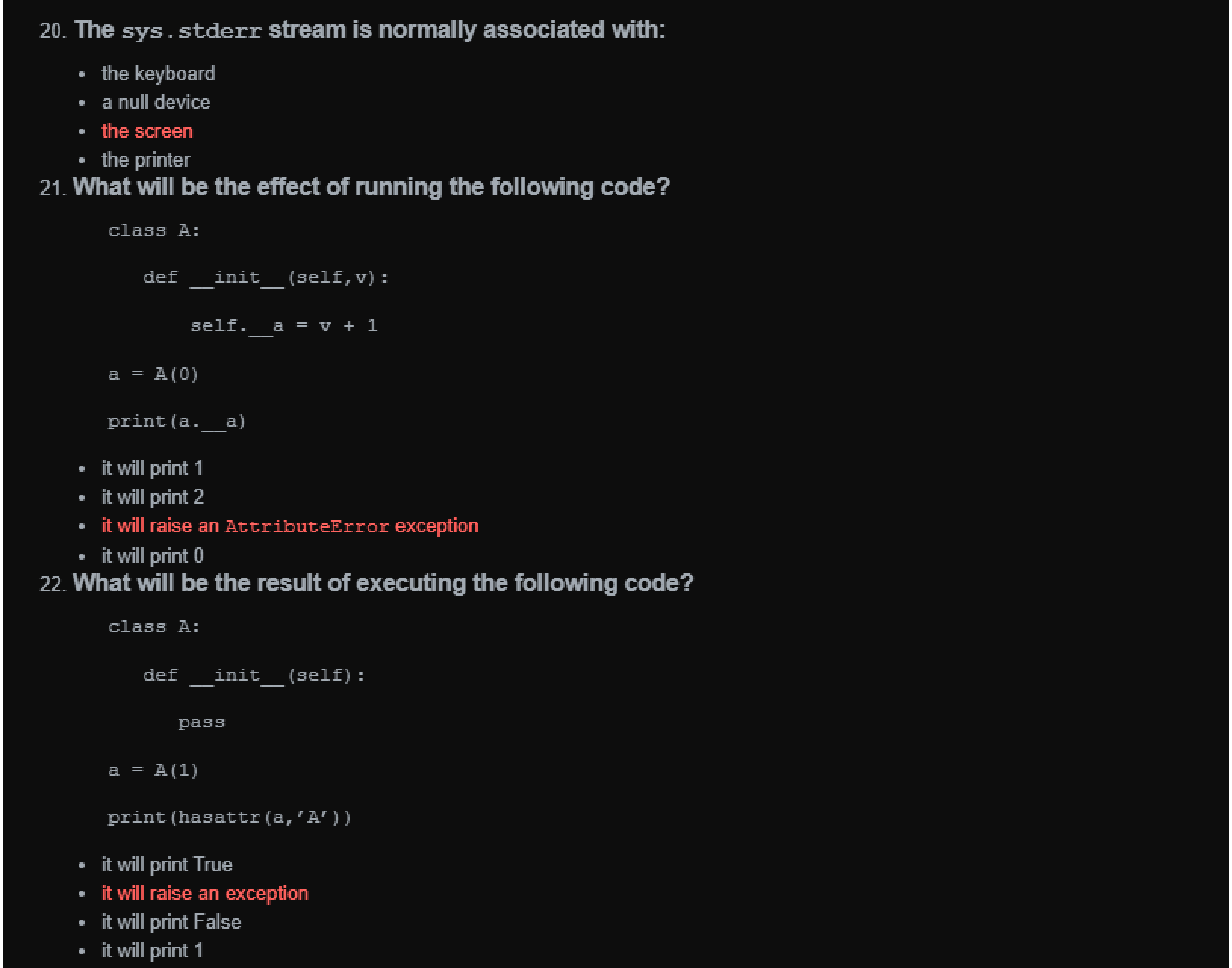
```
class A:

    A = 1

    def __init__(self):

        self.a = 0

print(hasattr(A,'a'))
```

- 1
- False
- 0
- True

19. **What will be the result of executing the following code?**

```
class A:

   pass

class B(A):

   pass

class C(B):

   pass

print(issubclass(A,C))
```

- it will print False
- it will print True
- it will print 1
- it will raise an exception



20. **The `sys.stderr` stream is normally associated with:**
    - the keyboard
    - a null device
    - [✓ the screen]
    - the printer
21. **What will be the effect of running the following code?**

```
class A:

   def __init__(self,v):

       self.__a = v + 1

a = A(0)

print(a.__a)
```

    - it will print 1
    - it will print 2
    - [✓ it will raise an `AttributeError` exception]
    - it will print 0
22. **What will be the result of executing the following code?**

```
class A:

   def __init__(self):

      pass

a = A(1)

print(hasattr(a,’A’))
```

    - it will print True
    - [✓ it will raise an exception]
    - it will print False
    - it will print 1

23. **What will be the result of executing the following code?**

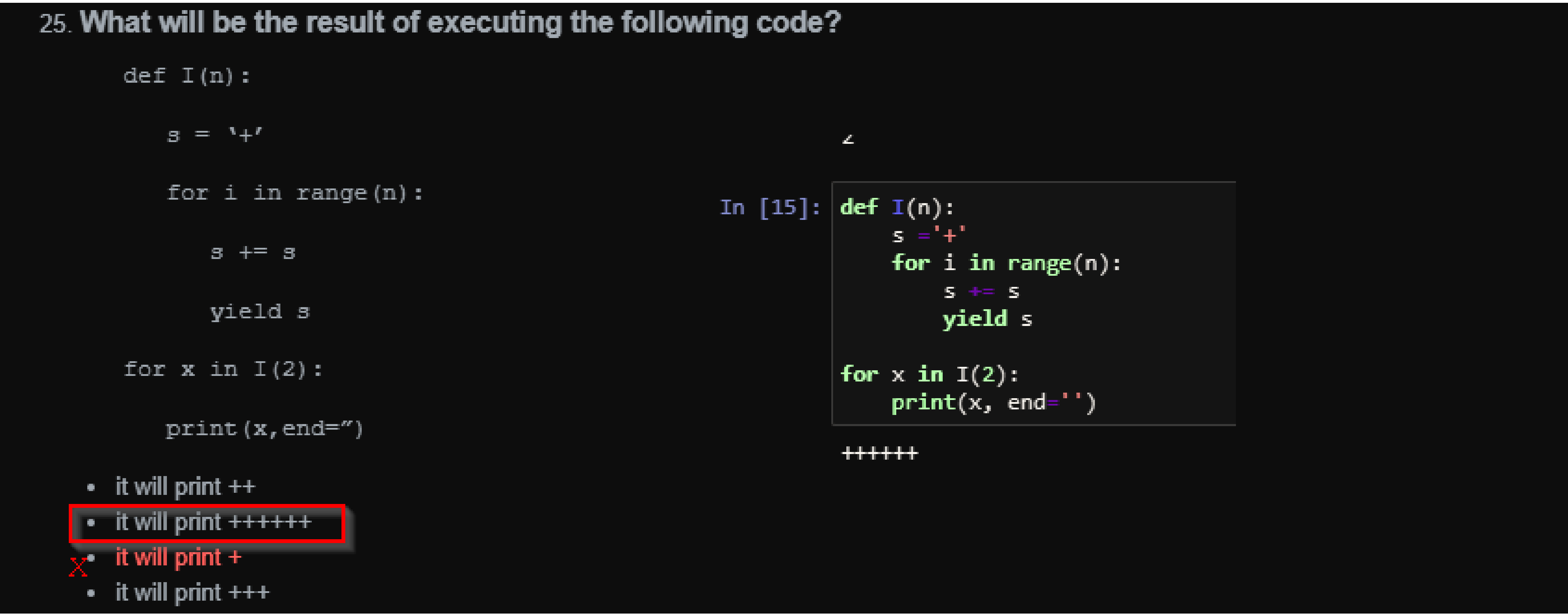
```
class A:

   def a(self):

      print('a')

class B:

   def a(self):

      print('b')

class C(B,A):

   def c(self):

      self.a()

o = C()

o.c()
```

- it will print c
- it will raise an exception
- [✓ it will print b]
- it will print a

24. **What will be the result of executing the following code?**

```
try:

   raise Exception(1,2,3)

except Exception as e:

   print(len(e.args))
```

- it will print 2
- it will print 1
- it will raise an unhandled exception
- [✓ it will print 3]



25. **What will be the result of executing the following code?**

```
def I(n):

    s = '+'

    for i in range(n):

        s += s

        yield s

for x in I(2):

    print(x,end='')
```

- it will print ++
- [✓ it will print ++++++]
- it will print +
- it will print +++

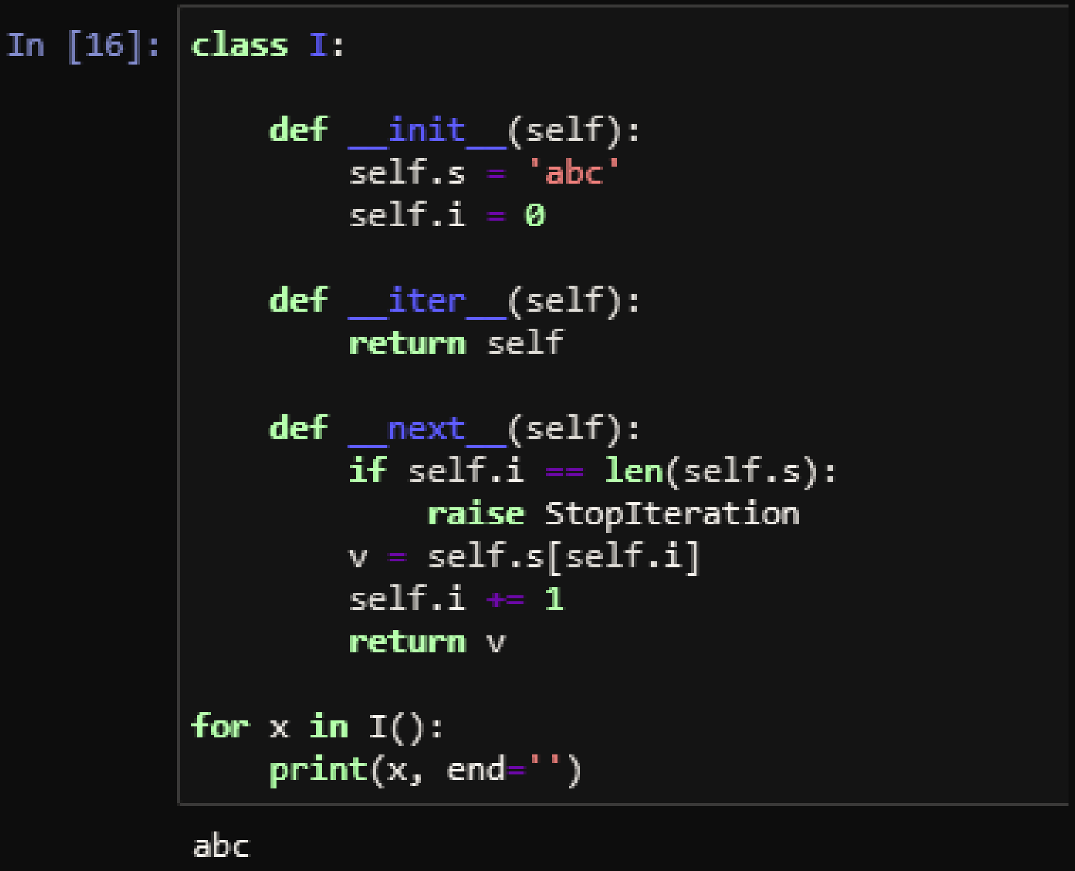
```
2
```

```
In [15]: def I(n):
             s ='+'
             for i in range(n):
                 s += s
                 yield s

         for x in I(2):
             print(x, end='')

         ++++++
```

• It will print +++

26. **What will be the result of executing the following code?**

```
class I:

   def __init__(self):

      self.s = 'abc'

      self.i = 0

   def __iter__(self):

      return self

   def __next__(self):

      if self.i == len(self.s):

         raise StopIteration

      v = self.s[self.i]

      self.i += 1

      return v

for x in I():

   print(x,end='')
```



```
In [16]: class I:

             def __init__(self):
                 self.s = 'abc'
                 self.i = 0

             def __iter__(self):
                 return self

             def __next__(self):
                 if self.i == len(self.s):
                     raise StopIteration
                 v = self.s[self.i]
                 self.i += 1
                 return v

         for x in I():
             print(x, end='')

         abc
```

- 012
- abc
- cba
- 210

27. **What will be the result of executing the following code?**

```
def o(p):

   def q():

      return ‘*’ * p

   return q

r = o(1)

s = o(2)

print(r() + s())
```

- it will print ****
- it will print **
- [✓ it will print ***]
- it will print *

28. **If s is a stream opened in read mode, the following line will:**

```
q = s.read(1)
```

- [✓ read 1 character from the stream]
- read 1 kilobyte from the stream
- read 1 line from the stream
- read 1 buffer from the stream

29. **Assuming that the open() invocation has gone successfully, the following snippet will:**

```
for x in open(‘file’,’rt’):

   print(x)
```

- read the file character by character
- cause an exception
- read the whole file at once
- [✓ read the file line by line]

30. **If you want to fill a byte array with data read in from a stream, you’d use:**

- [✓ the readinto() method]
- the read() method
- the readbytes() method
- the readfrom() method

PCAP – Programming Essentials in Python Quizzes Final Test Answers

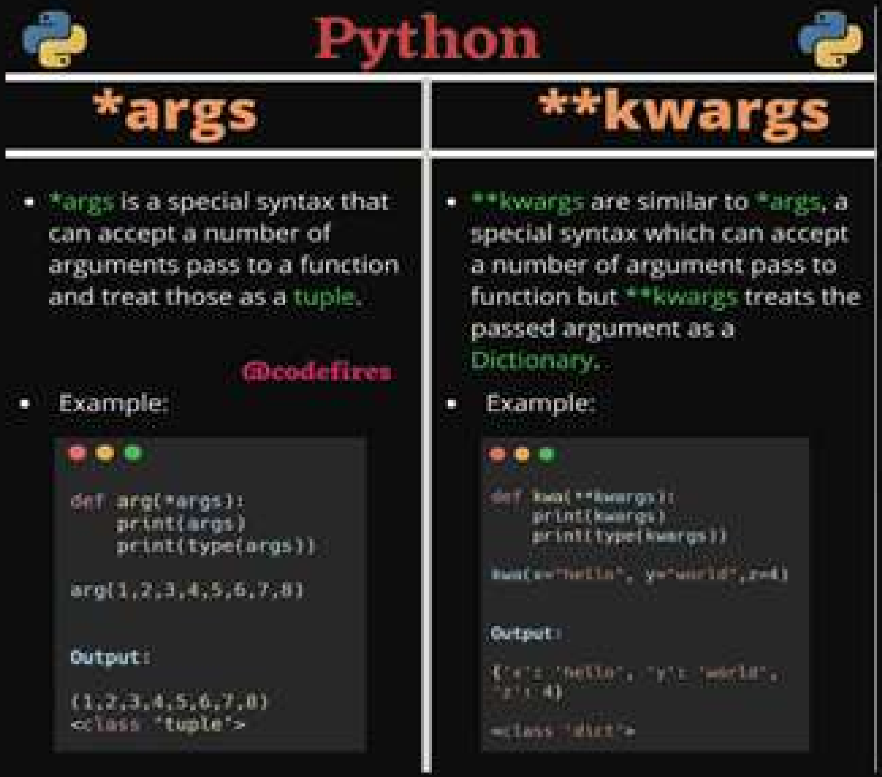
1. **The meaning of the keyword argument is determined by:**

   - its position within the argument list
   - its value
   - its connection with existing variables
   - the argument's name specified along with its value

2. **Which of the following sentences is true?**

```
str1 = 'string'

str2 = str1[:]
```

   - str1 and str2 are different (but equal) strings
   - str2 is longer than str1
   - ">str1 and str2 are different names of the same string
   - str1 is longer than str2



Python

*args

- *args is a special syntax that can accept a number of arguments pass to a function and treat those as a tuple.

@codefires

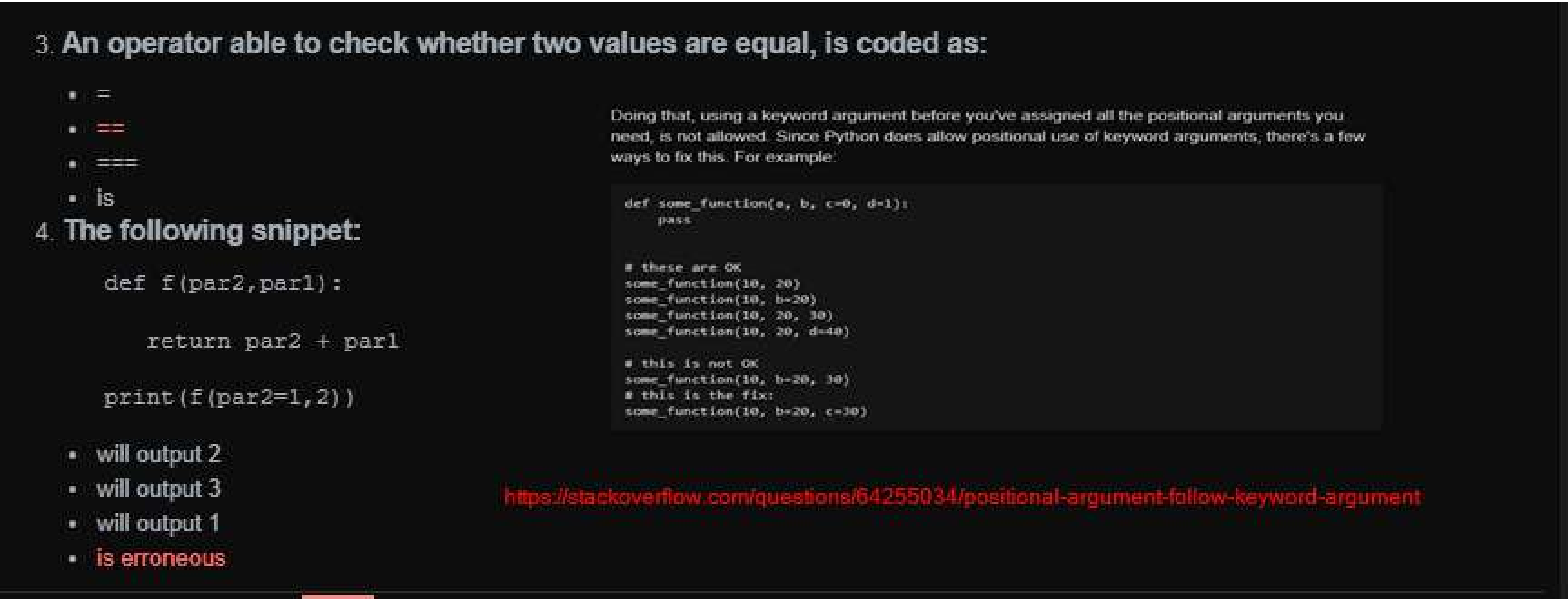
- Example:

```
def arg(*args):
    print(args)
    print(type(args))

arg(1,2,3,4,5,6,7,8)


Output:

(1,2,3,4,5,6,7,8)
<class 'tuple'>
```

**kwargs

- **kwargs are similar to *args, a special syntax which can accept a number of argument pass to function but **kwargs treats the passed argument as a Dictionary.
- Example:

```
def kwa(**kwargs):
    print(kwargs)
    print(type(kwargs))

kwa(x="hello", y="world",z=4)


Output:

{'x': 'hello', 'y': 'world',
 'z': 4}

<class 'dict'>
```



3. **An operator able to check whether two values are equal, is coded as:**

- =
- ==
- ===
- is

4. **The following snippet:**

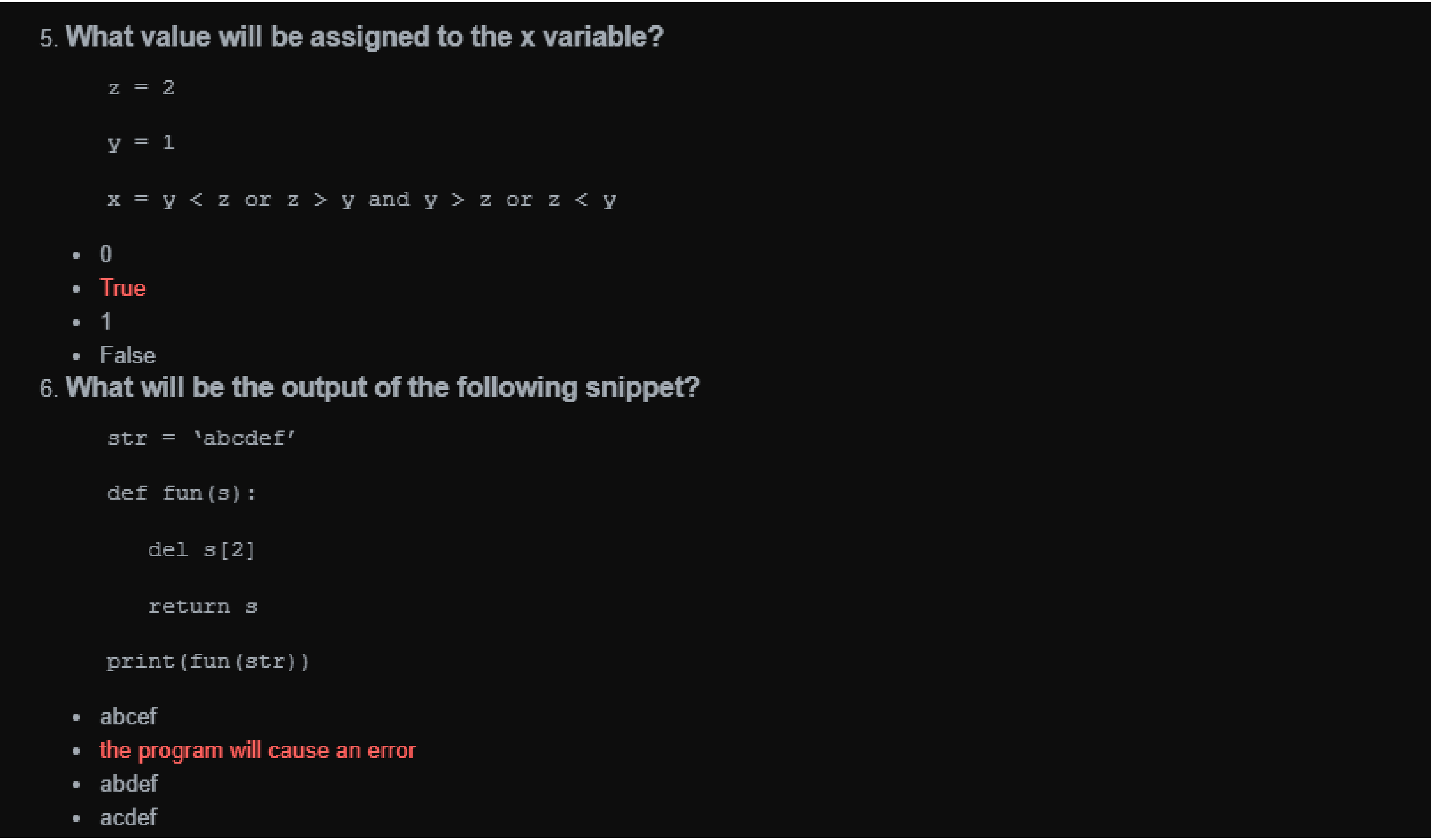
```
def f(par2,par1):

    return par2 + par1

print(f(par2=1,2))
```

- will output 2
- will output 3
- will output 1
- is erroneous

Doing that, using a keyword argument before you've assigned all the positional arguments you need, is not allowed. Since Python does allow positional use of keyword arguments, there's a few ways to fix this. For example:

```
def some_function(a, b, c=0, d=1):
    pass


# these are OK
some_function(10, 20)
some_function(10, b=20)
some_function(10, 20, 30)
some_function(10, 20, d=40)

# this is not OK
some_function(10, b=20, 30)
# this is the fix:
some_function(10, b=20, c=30)
```

https://stackoverflow.com/questions/64255034/positional-argument-follow-keyword-argument



5. **What value will be assigned to the x variable?**

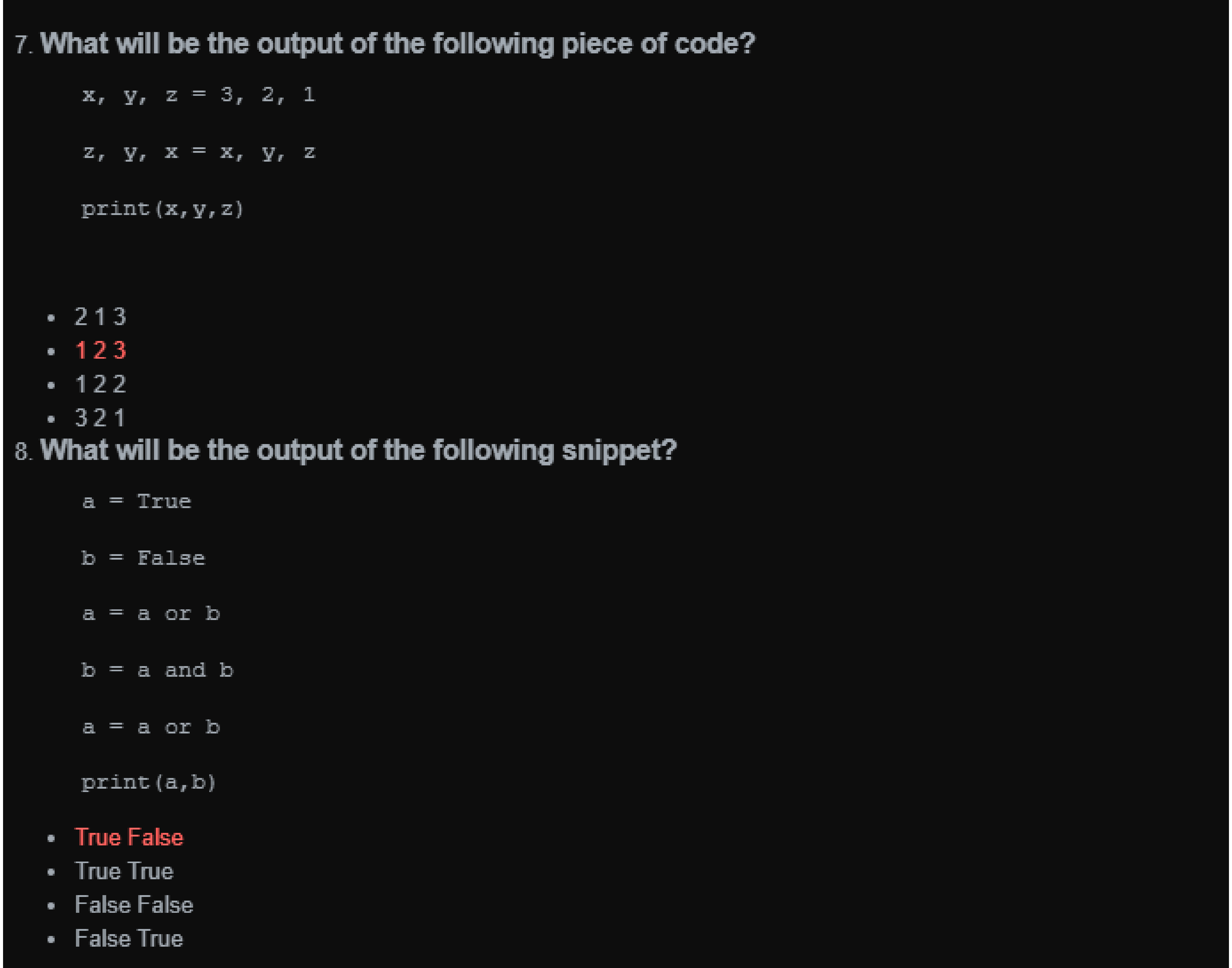
```
z = 2

y = 1

x = y < z or z > y and y > z or z < y
```

- 0
- [✓ True]
- 1
- False

6. **What will be the output of the following snippet?**

```
str = ‘abcdef’

def fun(s):

   del s[2]

   return s

print(fun(str))
```

- abcef
- [✓ the program will cause an error]
- abdef
- acdef



7. **What will be the output of the following piece of code?**

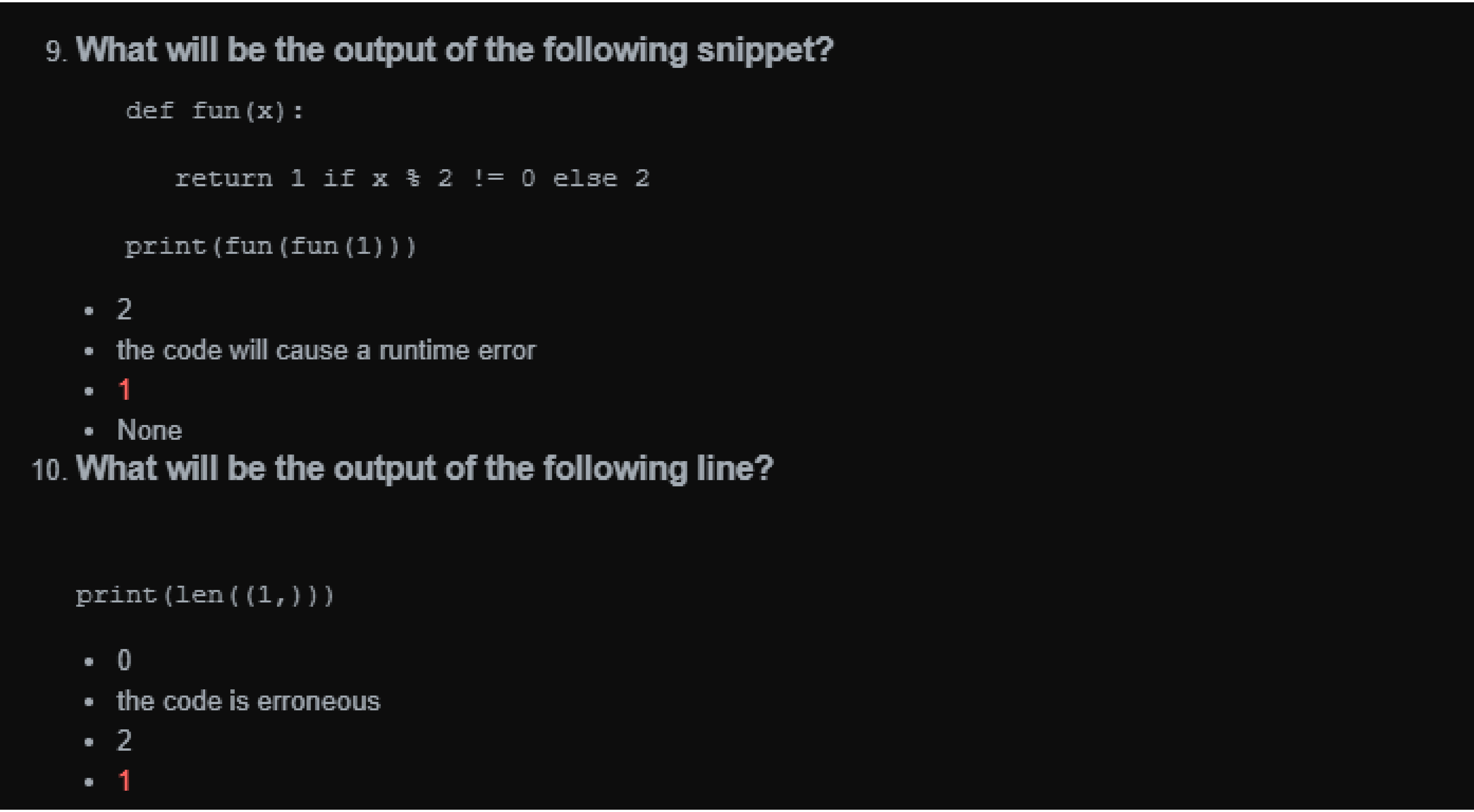
```
x, y, z = 3, 2, 1

z, y, x = x, y, z

print(x,y,z)
```

- 2 1 3
- [✓ 1 2 3]
- 1 2 2
- 3 2 1

8. **What will be the output of the following snippet?**

```
a = True

b = False

a = a or b

b = a and b

a = a or b

print(a,b)
```

- [✓ True False]
- True True
- False False
- False True



9. **What will be the output of the following snippet?**

```
def fun(x):

    return 1 if x % 2 != 0 else 2

print(fun(fun(1)))
```

- 2
- the code will cause a runtime error
- [✓ 1]
- None

10. **What will be the output of the following line?**

```
print(len((1,)))
```

- 0
- the code is erroneous
- 2
- [✓ 1]

11. **What will be the output of the following piece of code?**

```
d = { 1:0, 2:1, 3:2, 0:1 }

x = 0

for y in range(len(d)):

   x = d[x]

print(x)
```

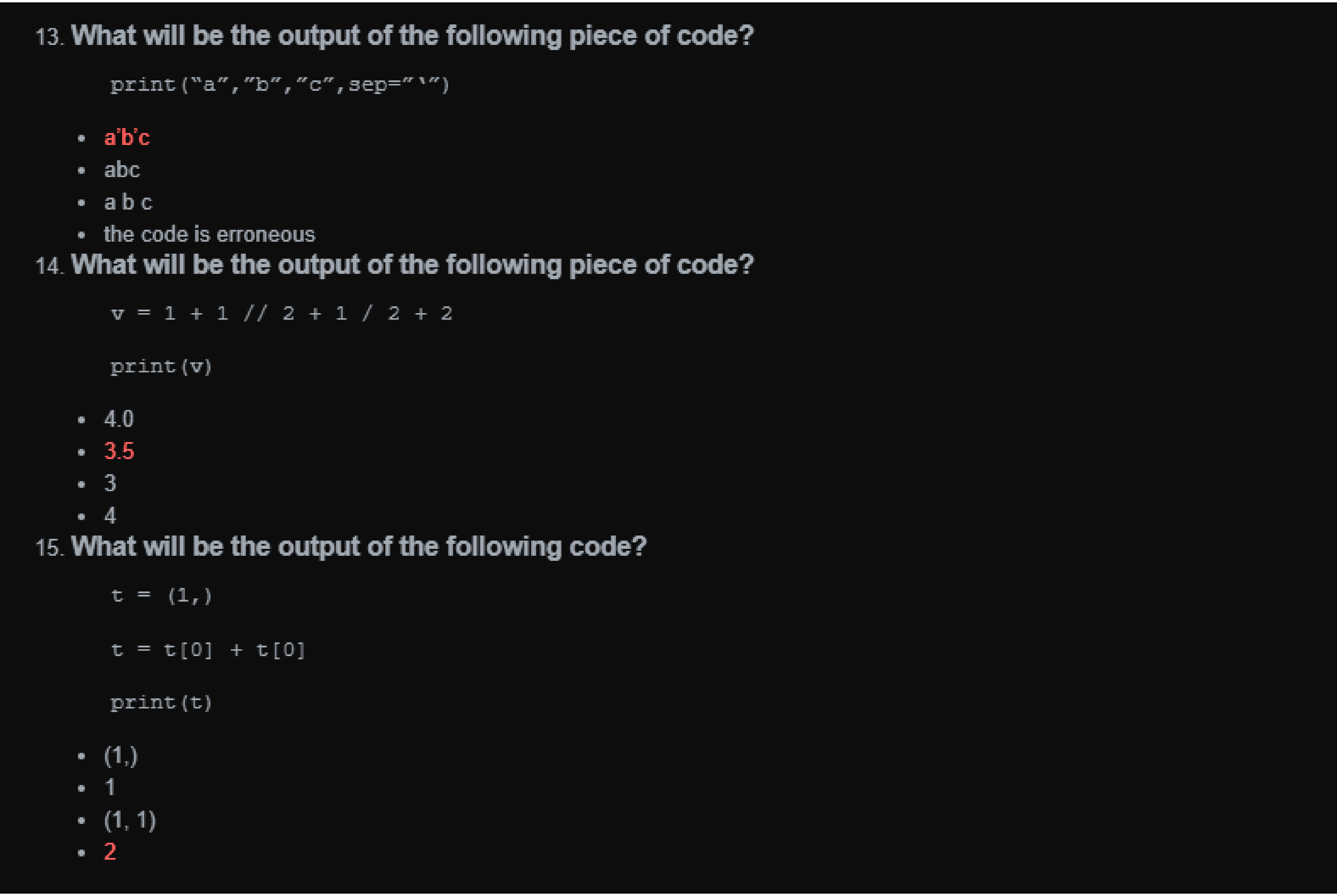
- the code will cause a runtime error
- 2
- [✓ 0]
- 1

12. **What will be the output of the following piece of code:**

```
y=input()

x=input()

print(x+y)
```

if the user enters two lines containing 1 and 2 respectively?

- [✓ 21]
- 12
- 2
- 3



13. **What will be the output of the following piece of code?**

```
print(“a”,”b”,”c”,sep=”‘”)
```

- a’b’c
- abc
- a b c
- the code is erroneous

14. **What will be the output of the following piece of code?**

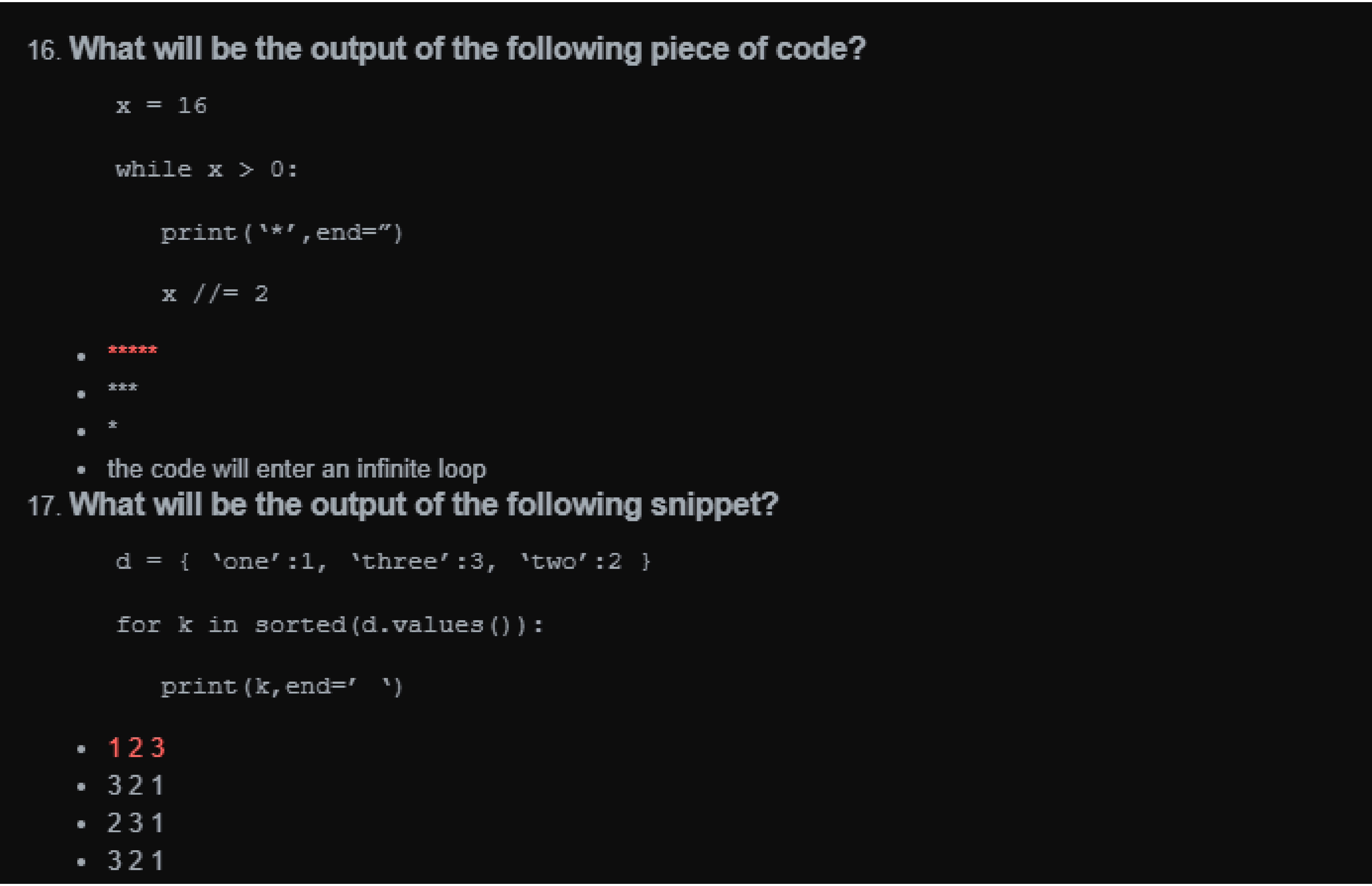
```
v = 1 + 1 // 2 + 1 / 2 + 2

print(v)
```

- 4.0
- 3.5
- 3
- 4

15. **What will be the output of the following code?**

```
t = (1,)

t = t[0] + t[0]

print(t)
```

- (1,)
- 1
- (1, 1)
- 2



16. **What will be the output of the following piece of code?**

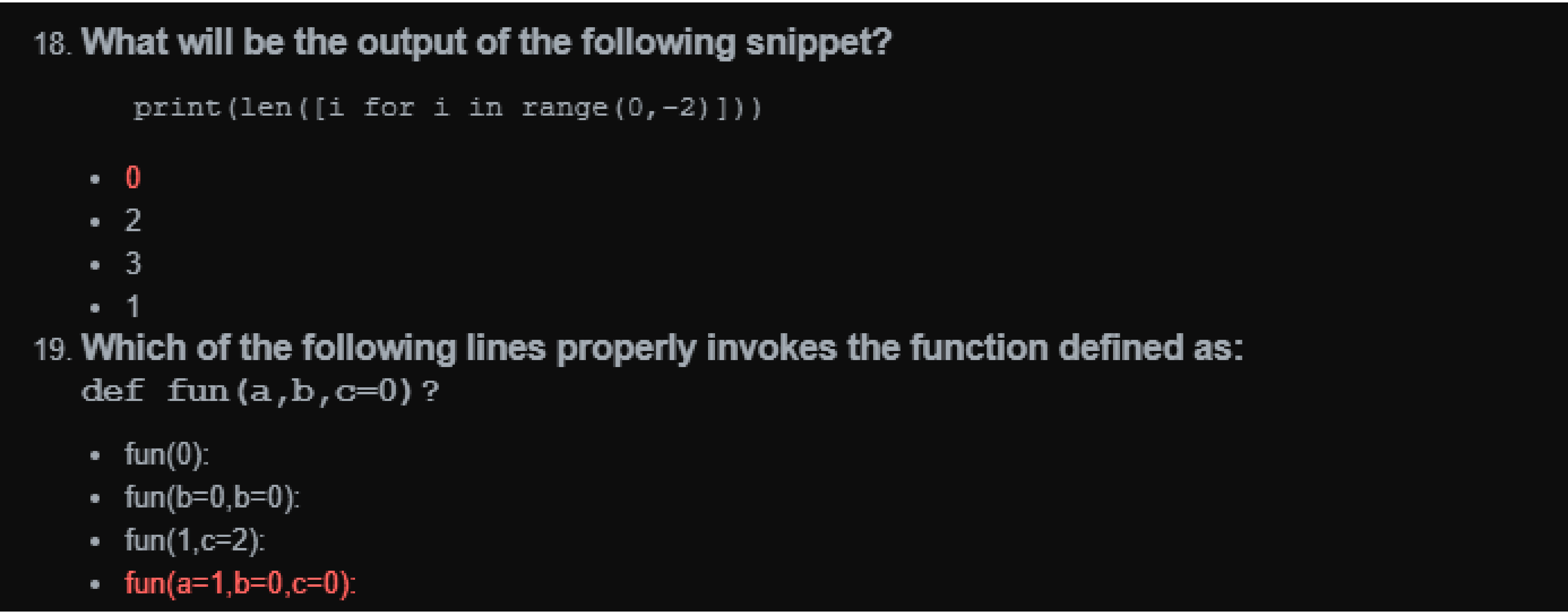
```
x = 16

while x > 0:

   print('*',end='')

   x //= 2
```

- *****
- ***
- *
- the code will enter an infinite loop

17. **What will be the output of the following snippet?**

```
d = { 'one':1, 'three':3, 'two':2 }

for k in sorted(d.values()):

   print(k,end=' ')
```

- 1 2 3
- 3 2 1
- 2 3 1
- 3 2 1



18. **What will be the output of the following snippet?**

```
print(len([i for i in range(0,-2)]))
```

- [✓ 0]
- 2
- 3
- 1

19. **Which of the following lines properly invokes the function defined as:** `def fun(a,b,c=0)`?

- fun(0):
- fun(b=0,b=0):
- fun(1,c=2):
- [✓ fun(a=1,b=0,c=0):]

20. **What will be the output of the following snippet?**

```
l = [1,2,3,4]

l = list(map(lambda x: 2*x,l))

print(l)
```

- 10
- the snippet will cause a runtime error
- 1 2 3 4
- [✓ 2 4 6 8]

21. **How many stars will the following snippet send to the console?**

```
i = 4

while i > 0 :

   i -= 2

   print(“*”)

   if i == 2:

      break

else:

   print(“*”)
```

- 2
- 0
- [✓ 1]
- the snippet will enter an infinite loop

22. **What will be the output of the following snippet?**

```
t = (1, 2, 3, 4)

t = t[-2:-1]

t = t[-1]

print(t)
```

- 33
- (3)
- [✓ 3]
- (3,)

23. **What will be the output of the following snippet?**

```
d = {}

d['2'] = [1,2]

d['1'] = [3,4]

for x in d.keys():

   print(d[x][1],end="")
```

- [✓ 24]
- 13
- 42
- 31

24. **What will be the output of the following snippet?**

```
def fun(d,k,v):

   d[k]=v

dc = {}

print(fun(dc,'1','v'))
```

- [✓ None]
- 1
- the snippet is erroneous
- v

25. **How many empty lines will the following snippet send to the console?**

```
l = [[c for c in range(r)] for r in range(3)]

for x in l:

   if len(x) < 2:

      print()
```

- 1
- 0
- [✓ 2]
- 3

26. **Knowing that the function named m() resides in the module named f, and the code contains the following import statement, choose the right way to invoke the function:**

```
from m import f
```

- [✓ the import statement is invalid]
- mod.fun()
- mod:fun()
- fun()

27. **The package directory/folder may contain a file intended to initialize the package. Its name is:**

- [✓ __init__.py]
- init.py
- __init.py__
- __init__.

28. **The folder created by Python used to store pyc files is named:**

- __pycfiles__
- __pyc__
- [✓ __pycache__]
- __cache__

29. **What will be the output of the following code, located in the file module.py?**

```
print(__name__)
```

- main
- __module.py__
- module.py
- [✓ __main__]

30. **If you want to tell your module's users that a particular variable should not be accessed directly, you may:**

- start its name with a capital letter
- use its number instead of its name
- [✓ start its name with _ or __]
- build its name with lowercase letters only

31. **If there is a `finally:` branch inside the `try:` block, we can say that:**

- it won't be executed if no exception is raised
- [✓ it will always be executed]
- branches is executed
- it will be executed when there is no else: branch

32. **What will be the output of the following snippet?**

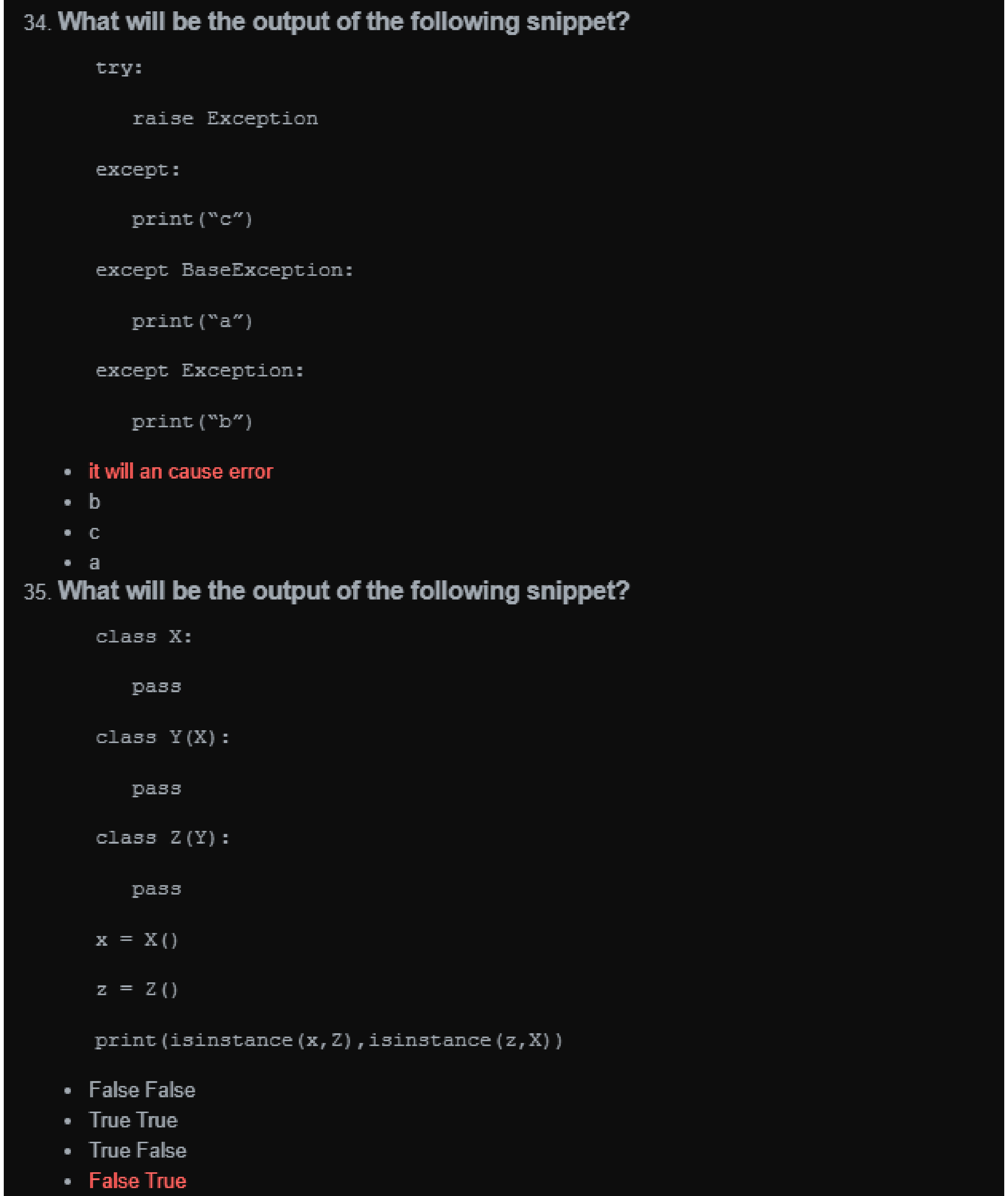
```
try:

   raise Exception

except BaseException:

   print(“a”,end=”)

else:

    print(“b”,end=”)

finally:

    print(“c”)
```

- a
- ab
- bc
- ac

33. **What will be the output of the following snippet?**

```
class A:

   def __init__(self,name):

      self.name = name

a = A(“class”)

print(a)
```

- a number
- a string ending with a long hexadecimal number
- class
- name



34. **What will be the output of the following snippet?**

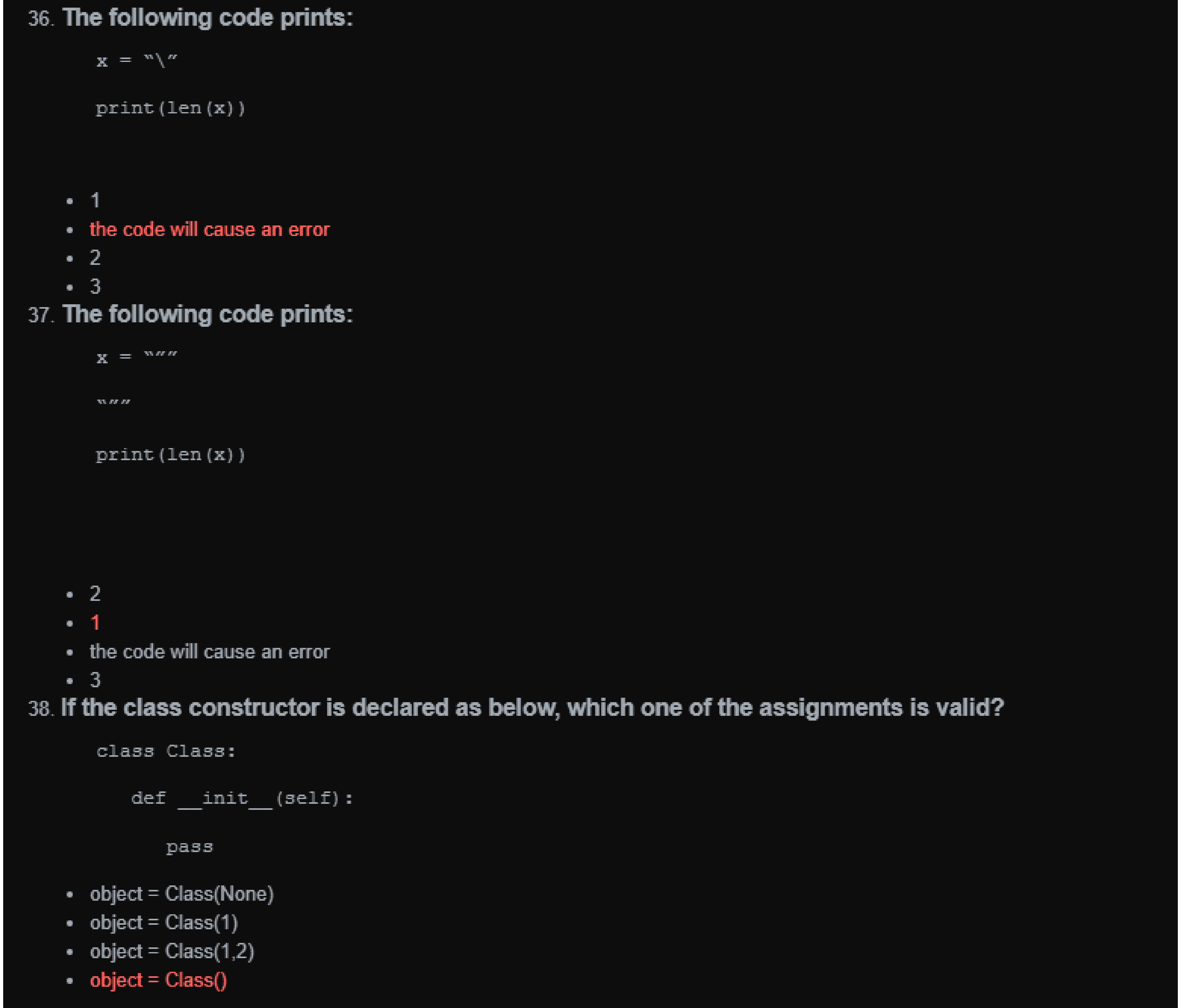
```
try:

   raise Exception

except:

   print(“c”)

except BaseException:

   print(“a”)

except Exception:

   print(“b”)
```

- [✓ it will an cause error]
- b
- c
- a

35. **What will be the output of the following snippet?**

```
class X:

   pass

class Y(X):

   pass

class Z(Y):

   pass

x = X()

z = Z()

print(isinstance(x,Z),isinstance(z,X))
```

- False False
- True True
- True False
- [✓ False True]



36. **The following code prints:**

```
x = "\"

print(len(x))
```

- 1
- [✓ the code will cause an error]
- 2
- 3

37. **The following code prints:**

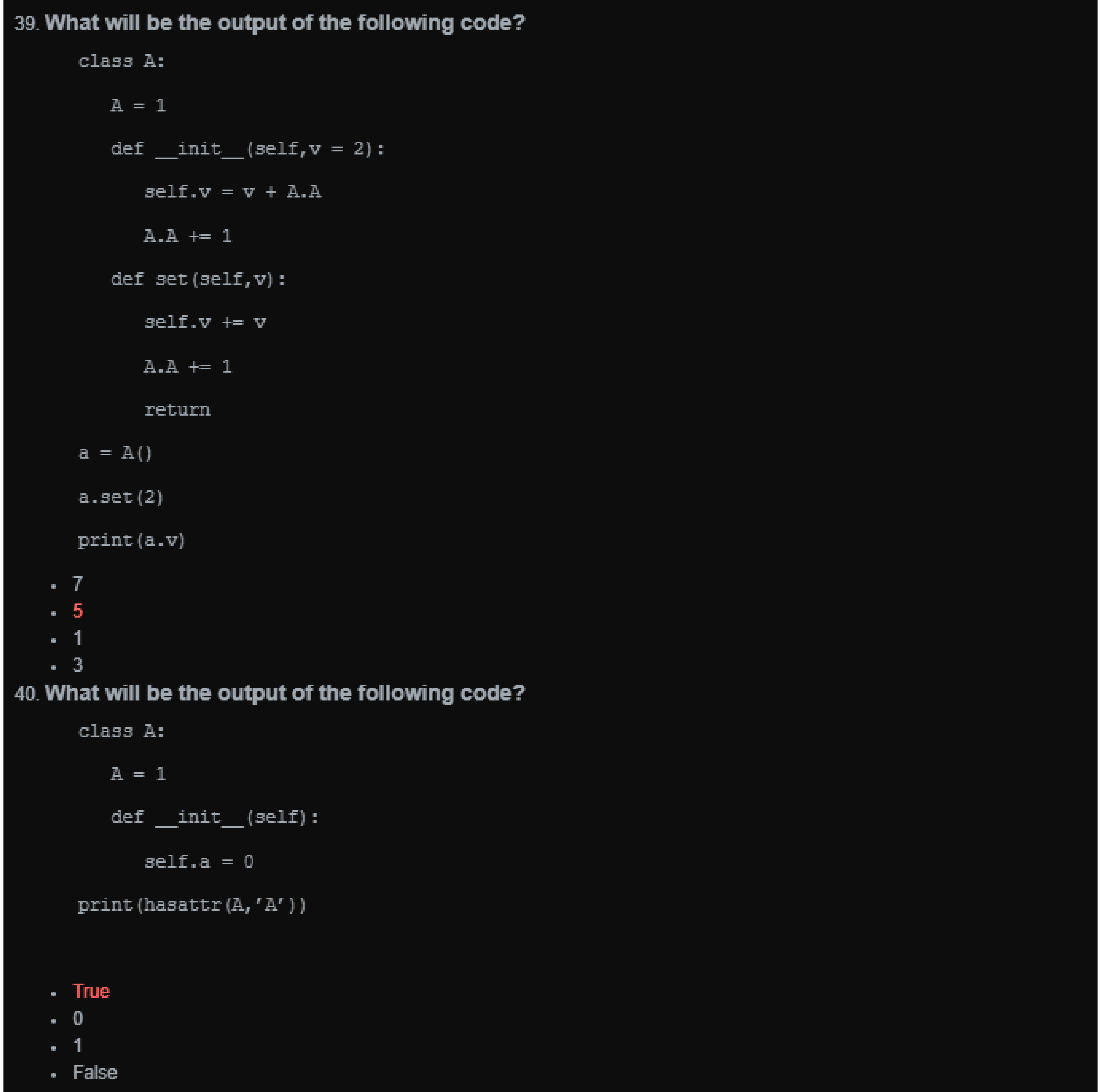
```
x = """

"""

print(len(x))
```

- 2
- [✓ 1]
- the code will cause an error
- 3

38. **If the class constructor is declared as below, which one of the assignments is valid?**

```
class Class:

   def __init__(self):

      pass
```

- object = Class(None)
- object = Class(1)
- object = Class(1,2)
- [✓ object = Class()]



39. **What will be the output of the following code?**

```
class A:

    A = 1

    def __init__(self,v = 2):

        self.v = v + A.A

        A.A += 1

    def set(self,v):

        self.v += v

        A.A += 1

        return

a = A()

a.set(2)

print(a.v)
```

- 7
- [✓ 5]
- 1
- 3

40. **What will be the output of the following code?**

```
class A:

    A = 1

    def __init__(self):

        self.a = 0

print(hasattr(A,'A'))
```

- [✓ True]
- 0
- 1
- False

41. **What will be the result of executing the following code?**

```
class A:

   pass

class B:

   pass

class C(A,B):

   pass

print(issubclass(C,A) and issubclass(C,B))
```

- [✓ it will print True]
- it will raise an exception
- it will print an empty line
- it will print False

42. **The `sys.stdout` stream is normally associated with:**

- [✓ the screen]
- a null device
- the keyboard
- the printer

43. **What will be the effect of running the following code?**

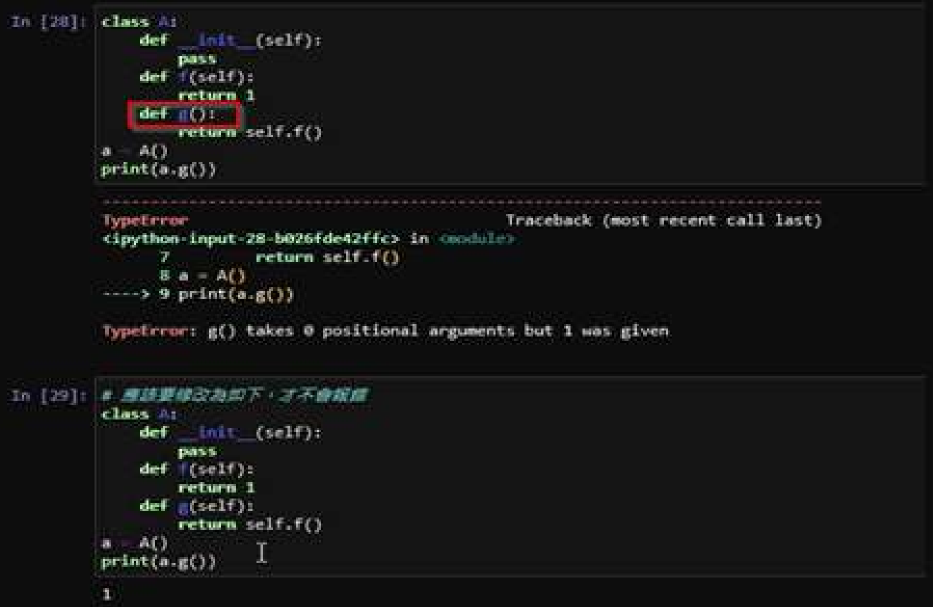
```
class A:

   def __init__(self,v):

   self._a = v + 1

a = A(0)

print(a._a)
```

- it will print 0
- [✓ it will print 1]
- it will print 2
- it will raise an AttributeError exception

44. **What will be the result of executing the following code?**

```
class A:
   def __init__(self):
      pass
   def f(self):
      return 1
   def g():
      return self.f()
a = A()
print(a.g())
```

- it will print 0
- it will print True
- it will print 1
- it will raise an exception



```
In [28]: class A:
             def __init__(self):
                 pass
             def f(self):
                 return 1
             def g():
                 return self.f()
         a = A()
         print(a.g())

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         <ipython-input-28-b026fde42ffc> in <module>
               7         return self.f()
               8 a = A()
         ----> 9 print(a.g())

         TypeError: g() takes 0 positional arguments but 1 was given

In [29]: # 應該要修改為如下，才不會報錯
         class A:
             def __init__(self):
                 pass
             def f(self):
                 return 1
             def g(self):
                 return self.f()
         a = A()
         print(a.g())

         1
```

45. **What will be the result of executing the following code?**

```
class A:
   def a(self):
     print('a')
class B:
   def a(self):
      print('b')
class C(A,B):
   def c(self):
      self.a()
o = C()
o.c()
```

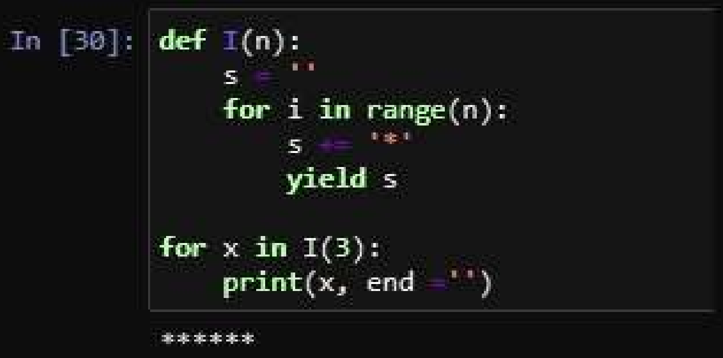
- it will print b
- it will print a
- it will raise an exception
- it will print c

46. **The Exception class contains a property named args, and it is a:**

- string
- [✓ tuple]
- list
- dictionary

47. **What will be the result of executing the following code?**

```
def I(n):

   s = ”

   for i in range(n):

      s += ‘*’

      yield s

   for x in I(3):

      print(x,end=”)
```

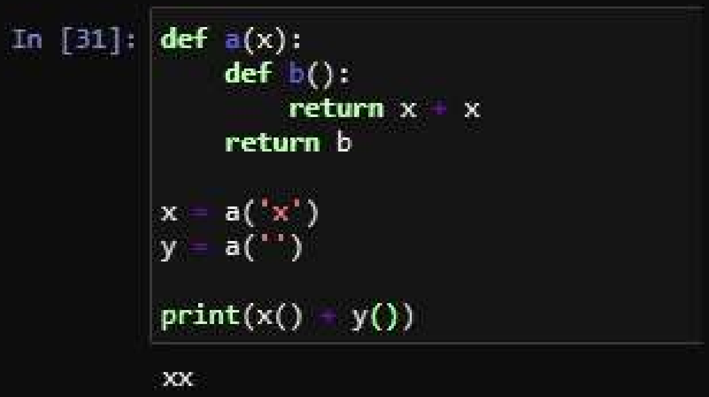


```
In [30]: def I(n):
             s = ''
             for i in range(n):
                 s += '*'
                 yield s

         for x in I(3):
             print(x, end ='')

         ******
```

- it will print ***
- it will print ****
- it will print *
- [✓ it will print ******]

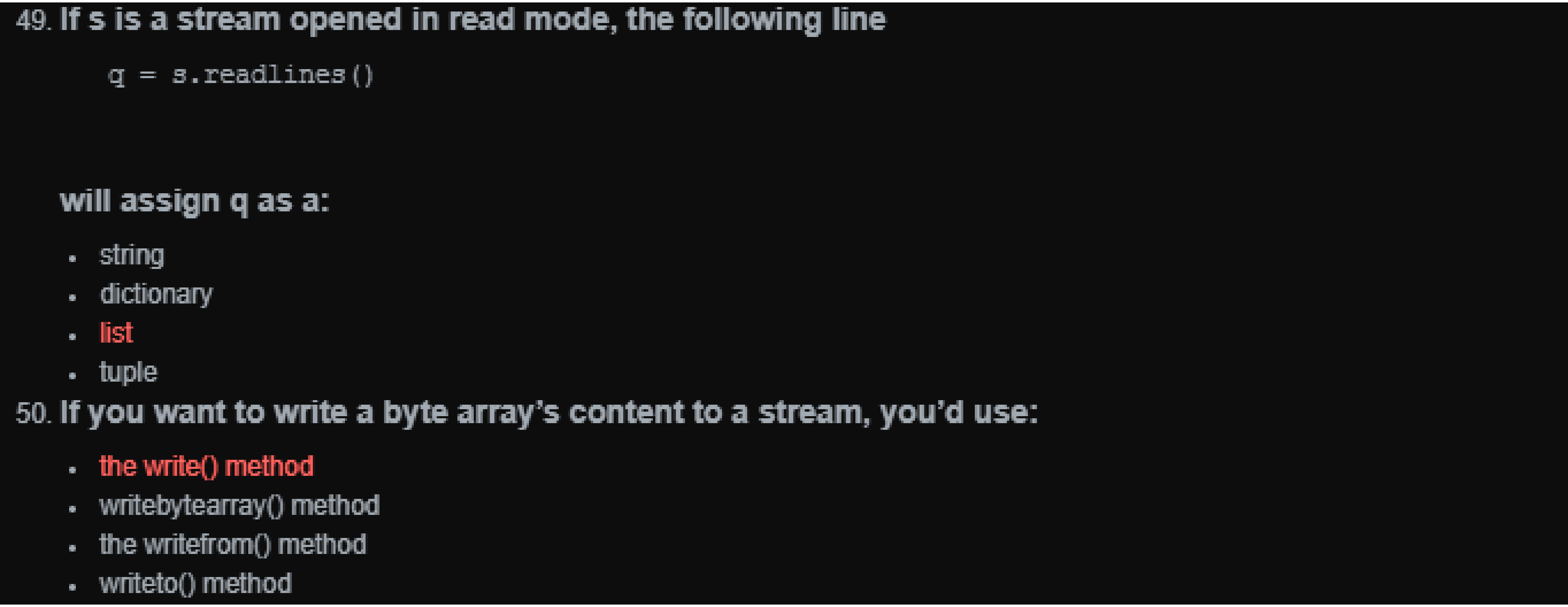
48. **What will be the result of executing the following code?**

```
def a(x):

   def b():

      return x + x

   return b

x = a(‘x’)

y = a(”)

print(x() + y())
```



```
In [31]: def a(x):
             def b():
                 return x + x
             return b

         x = a('x')
         y = a('')

         print(x() + y())

         xx
```

- it will print xxxxxx
- it will print x
- [✓ it will print xx]
- it will print xxxx



49. **If s is a stream opened in read mode, the following line**

```
q = s.readlines()
```

will assign q as a:

- string
- dictionary
- [✓ list]
- tuple

50. **If you want to write a byte array's content to a stream, you'd use:**

- [✓ the write() method]
- writebytearray() method
- the writefrom() method
- writeto() method