PCEP-30-02 Q&As

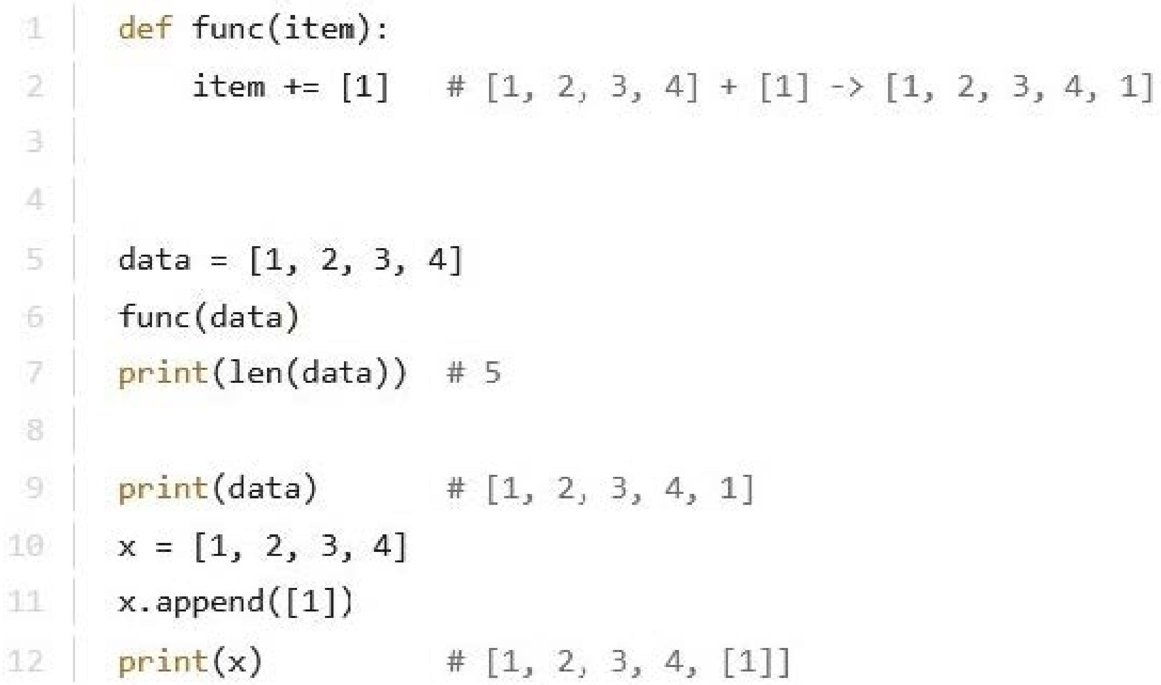
PCEP - Certified Entry-Level Python Programmer

Pass Python Institute PCEP-30-02 Exam with 100% Guarantee

Following Questions and Answers are all new published by Python Institute Official Exam Center

QUESTION 1

What is the expected output of the following code?



```
def func(item):
    item += [1]   # [1, 2, 3, 4] + [1] -> [1, 2, 3, 4, 1]


data = [1, 2, 3, 4]
func(data)
print(len(data))  # 5

print(data)       # [1, 2, 3, 4, 1]
x = [1, 2, 3, 4]
x.append([1])
print(x)          # [1, 2, 3, 4, [1]]
```

A. 5

B. 4

C. 2

D. The code is erroneous.

Correct Answer: A

QUESTION 2

Which of the following operators can be used with strings?

1) +

2) *

3) 4) in

A. 1, 2, 3

B. 1, 2, 4

C. 1, 2, 3, 4

D. 1, 2

Correct Answer: B

QUESTION 3

What is true about exceptions and debugging? (Select two answers.)

A. A tool that allows you to precisely trace program execution is called a debugger.

B. If some Python code is executed without errors, this proves that there are no errors in it.

C. One try-except block may contain more than one except branch.

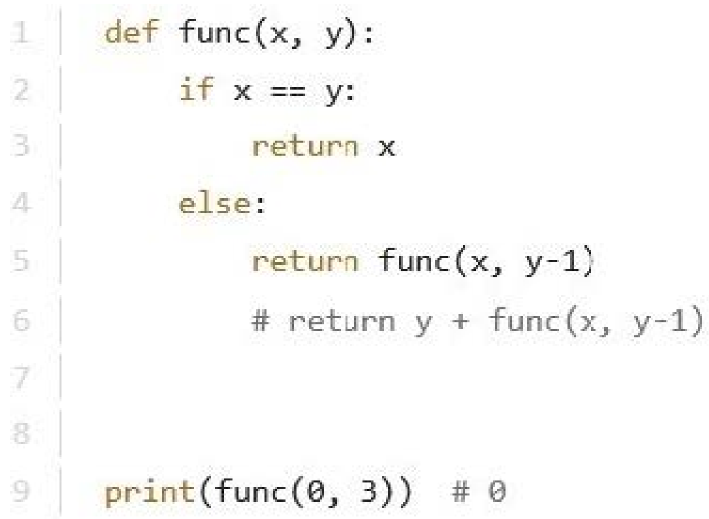
D. The default (anonymous) except branch cannot be the last branch in the try-except block.

Correct Answer: AC

Explanation: Exceptions and debugging are two important concepts in Python programming that are related to handling and preventing errors. Exceptions are errors that occur when the code cannot be executed properly, such as syntax errors, type errors, index errors, etc. Debugging is the process of finding and fixing errors in the code, using various tools and techniques. Some of the facts about exceptions and debugging are: A tool that allows you to precisely trace program execution is called a debugger. A debugger is a program that can run another program step by step, inspect the values of variables, set breakpoints, evaluate expressions, etc. A debugger can help you find the source and cause of an error, and test possible solutions. Python has a built-in debugger module called pdb, which can be used from the command line or within the code. There are also other third-party debuggers available for Python, such as PyCharm, Visual Studio Code, etc12 If some Python code is executed without errors, this does not prove that there are no errors in it. It only means that the code did not encounter any exceptions that would stop the execution. However, the code may still have logical errors, which are errors that cause the code to produce incorrect or unexpected results. For example, if you write a function that is supposed to calculate the area of a circle, but you use the wrong formula, the code may run without errors, but it will give you the wrong answer. Logical errors are harder to detect and debug than syntax or runtime errors, because they do not generate any error messages. You have to test the code with different inputs and outputs, and compare them with the expected results34 One try-except block may contain more than one except branch. A try-except block is a way of handling exceptions in Python, by using the keywords try and except. The try block contains the code that may raise an exception, and the except block contains the code that will execute if an exception occurs. You can have multiple except blocks for different types of exceptions, or for different actions to take. For example, you can write a try-except block like this: try: # some code that may raise an exception except ValueError: # handle the ValueError exception except ZeroDivisionError: # handle the ZeroDivisionError exception except: # handle any other exception This way, you can customize the error handling for different situations, and provide more informative messages or alternative solutions5 The default (anonymous) except branch can be the last branch in the try-except block. The default except branch is the one that does not specify any exception type, and it will catch any exception that is not handled by the previous except branches. The default except branch can be the last branch in the try-except block, but it cannot be the first or the only branch. For example, you can write a try- except block like this: try: # some code that may raise an exception except ValueError: # handle the ValueError exception except: # handle any other exception This is a valid try-except block, and the default except branch will be the last branch. However, you cannot write a try-except block like this: try: # some code that may raise an exception except: # handle any exception This is an invalid try-except block, because the default except branch is the only branch, and it will catch all exceptions, even those that are not errors, such as KeyboardInterrupt or SystemExit. This is considered a bad practice, because it may hide or ignore important exceptions that should be handled differently or propagated further. Therefore, you should always specify the exception types that you want to handle, and use the default except branch only as a last resort5 Therefore, the correct answers are A. A tool that allows you to precisely trace program execution is called a debugger. and C. One try-except block may contain more than one except branch. Reference: Python Debugger ?Python pdb - GeeksforGeeksHow can I see the details of an exception in Python\\'s debugger?Python Debugging (fixing problems)Python - start interactive debugger when exception would be otherwise thrownPython Try Except [Error Handling and Debugging -- Programming with Python for Engineers]

QUESTION 4

What is the output of the following snippet?



```
def func(x, y):
    if x == y:
        return x
    else:
        return func(x, y-1)
        # return y + func(x, y-1)


print(func(0, 3))  # 0
```

A. The snippet will cause a runtime error.

B. 0

C. 1

D. 2

Correct Answer: B

QUESTION 5

The pyc file contains ...

A. a Python interpreter.

B. Python source code.

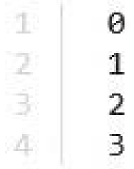
C. compiled Python bytecode.

D. a Python compiler.

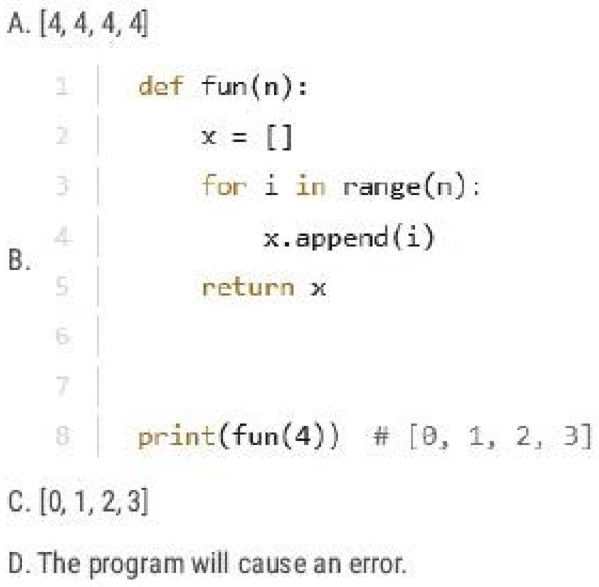
Correct Answer: C

QUESTION 6

What is the expected output of the following code?



```
0
1
2
3
```



A. [4, 4, 4, 4]

B.

```
def fun(n):
    x = []
    for i in range(n):
        x.append(i)
    return x


print(fun(4))  # [0, 1, 2, 3]
```

C. [0, 1, 2, 3]

D. The program will cause an error.

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

QUESTION 7

The following statement ... assert x == 0

A. will stop the program if x is equal to 0.

B. has no effect.

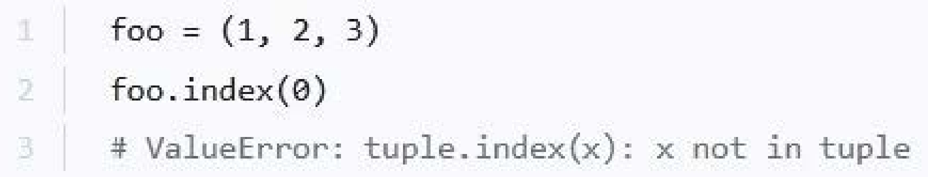
C. is erroneous.

D. will stop the program if x is not equal to 0.

Correct Answer: D

QUESTION 8

What is the expected behavior of the following program?



```
foo = (1, 2, 3)
foo.index(0)
# ValueError: tuple.index(x): x not in tuple
```

A. The program will cause a AttributeError exception.

B. The program will cause a SyntaxError exception.

C. The program will output 1 to the screen.

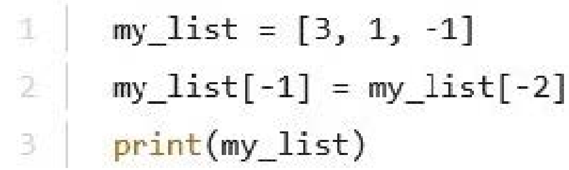
D. The program will cause a ValueError exception.

E. The program will cause a TypeError exception.

Correct Answer: D

QUESTION 9

What is the output of the following code?



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

A. [1, 1, 1]

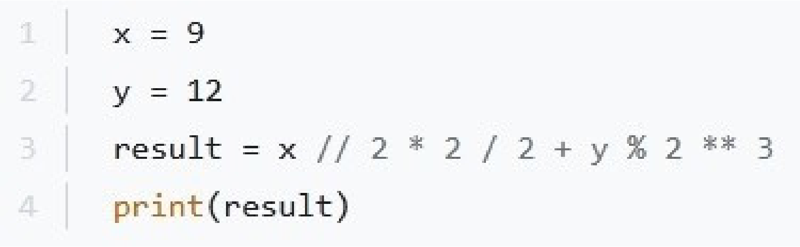
B. [3, -1, 1]

C. [3, 1, 1]

Correct Answer: C

QUESTION 10

What is the expected output of the following code?



```
x = 9
y = 12
result = x // 2 * 2 / 2 + y % 2 ** 3
print(result)
```

A. 8

B. 7.0

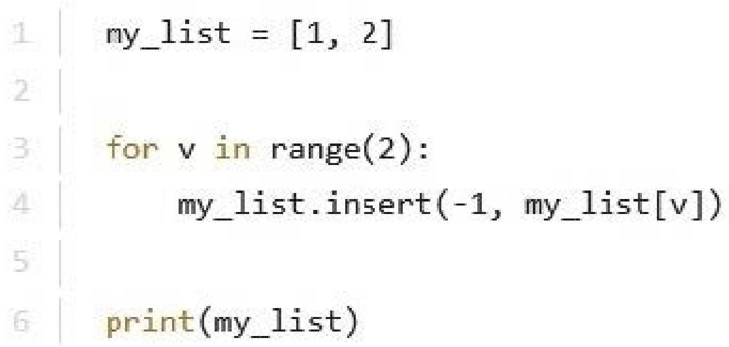
C. 8.0

D. 9.0

Correct Answer: C

QUESTION 11

What is the output of the following snippet?



```
my_list = [1, 2]

for v in range(2):
    my_list.insert(-1, my_list[v])

print(my_list)
```

A. [1, 1, 2, 2]

B. [1, 1, 1, 2]

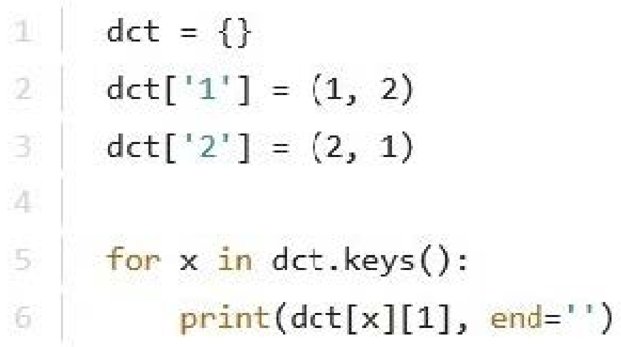
C. [1, 2, 1, 2]

D. [1, 2, 2, 2]

Correct Answer: B

QUESTION 12

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='')
```

A. 12

B. (2, 1)

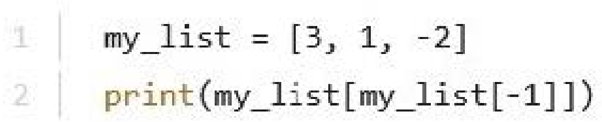
C. (1, 2)

D. 21

Correct Answer: D

QUESTION 13

What is the output of the following snippet?



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

A. -2

B. 3

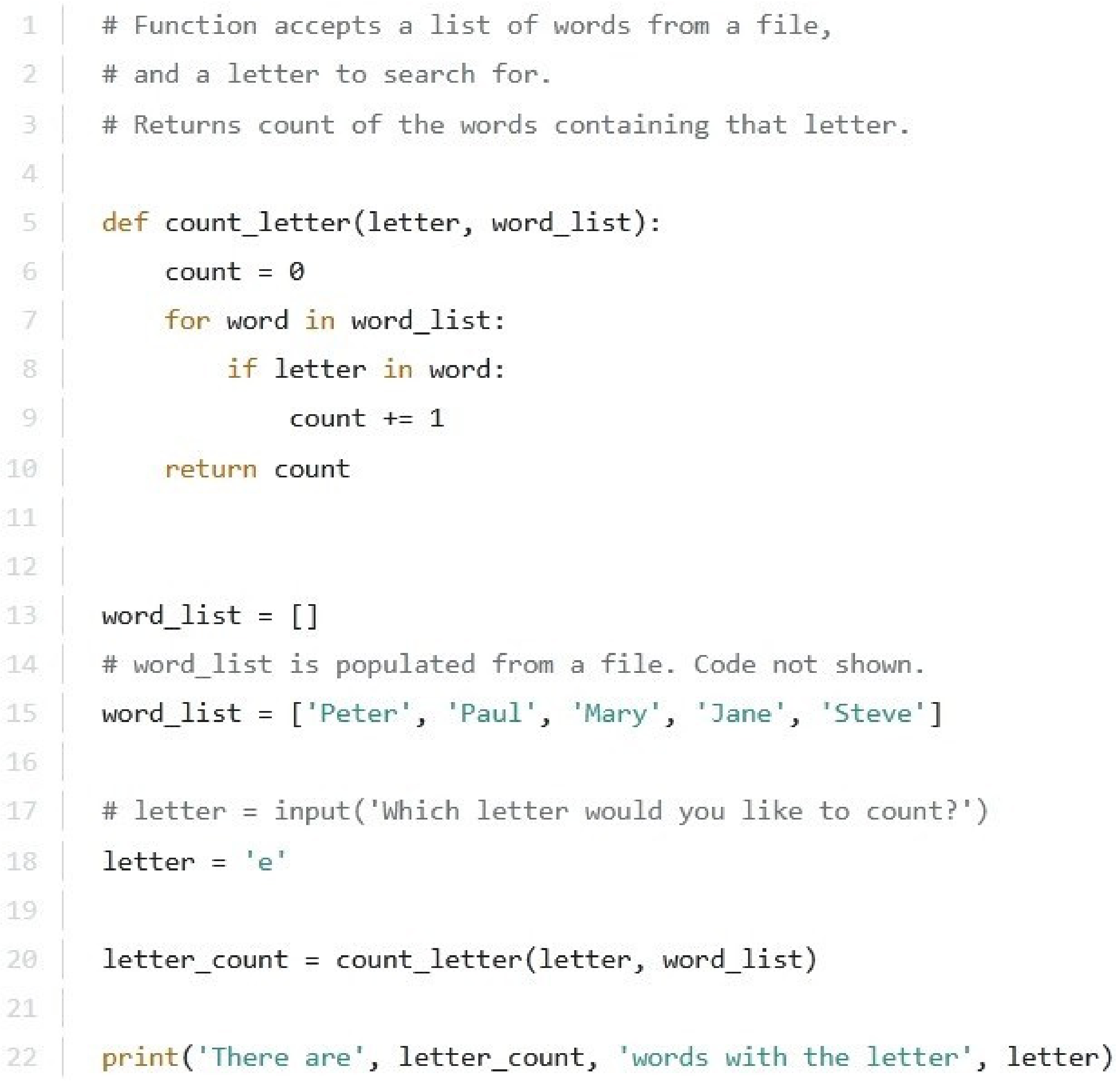
C. -1

D. 1

Correct Answer: D

QUESTION 14

The ABC company needs a way to find the count of particular letters in their publications to ensure that there is a good balance. It seems that there have been complaints about overuse of the letter e. You need to create a function to meet the requirements.



```
# Function accepts a list of words from a file,
# and a letter to search for.
# Returns count of the words containing that letter.

def count_letter(letter, word_list):
    count = 0
    for word in word_list:
        if letter in word:
            count += 1
    return count


word_list = []
# word_list is populated from a file. Code not shown.
word_list = ['Peter', 'Paul', 'Mary', 'Jane', 'Steve']

# letter = input('Which letter would you like to count?')
letter = 'e'

letter_count = count_letter(letter, word_list)

print('There are', letter_count, 'words with the letter', letter)
```

What would you insert instead of ??? and ??? ?

A. 1 | word in word_list 2 | word in letter

B. 1 | word_list in word 2 | letter in word

C. 1 | word in word_list 2 | letter in word

D. 1 | word is word_list 2 | letter in word

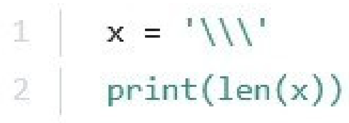
E. 1 | word_list in word 2 | word in letter

F. 1 | word in word_list 2 | letter is word

Correct Answer: C

QUESTION 15

What is the expected output of the following code?



```
x = '\\\'
print(len(x))
```

A. The code is erroneous.

B. 2

C. 3

D. 1

Correct Answer: A

Latest PCEP-30-02 Dumps PCEP-30-02 VCE Dumps PCEP-30-02 Practice Test