1. Which of the following functions is a built-in function in python?  
a) seed()  
b) sqrt()  
c) factorial()  
d) print()

Answer: d  
Explanation: The function seed is a function which is present in the random module. The functions sqrt and factorial are a part of the math module. The print function is a built-in function which prints a value directly to the system output.

2. What will be the output of the following Python expression?

round(4.576)

a) 4.5  
b) 5  
c) 4  
d) 4.6

Answer: b  
Explanation: This is a built-in function which rounds a number to give precision in decimal digits. In the above case, since the number of decimal places has not been specified, the decimal number is rounded off to a whole number. Hence the output will be 5.

3. The function pow(x,y,z) is evaluated as:  
a) (x\*\*y)\*\*z  
b) (x\*\*y) / z  
c) (x\*\*y) % z  
d) (x\*\*y)\*z

Answer: c  
Explanation: The built-in function pow() can accept two or three arguments. When it takes in two arguments, they are evaluated as x\*\*y. When it takes in three arguments, they are evaluated as (x\*\*y)%z.

4. What will be the output of the following Python function?

all([2,4,0,6])

a) Error  
b) True  
c) False  
d) 0

Answer: c  
Explanation: The function all returns false if any one of the elements of the iterable is zero and true if all the elements of the iterable are non zero. Hence the output of this function will be false.

5. What will be the output of the following Python expression?

round(4.5676,2)?

a) 4.5  
b) 4.6  
c) 4.57  
d) 4.56

Answer: c  
Explanation: The function round is used to round off the given decimal number to the specified decimal places. In this case, the number should be rounded off to two decimal places. Hence the output will be 4.57.

6. What will be the output of the following Python function?

any([2>8, 4>2, 1>2])

a) Error  
b) True  
c) False  
d) 4>2

Answer: b  
Explanation: The built-in function any() returns true if any or more of the elements of the iterable is true (non zero), If all the elements are zero, it returns false.

7. What will be the output of the following Python function?

**import** math

abs(math.sqrt(25))

a) Error  
b) -5  
c) 5  
d) 5.0

Answer: d  
Explanation: The abs() function prints the absolute value of the argument passed. For example: abs(-5)=5. Hence, in this case we get abs(5.0)=5.0.

8. What will be the output of the following Python function?

sum(2,4,6)

sum([1,2,3])

a) Error, 6  
b) 12, Error  
c) 12, 6  
d) Error, Error

Answer: a  
Explanation: The first function will result in an error because the function sum() is used to find the sum of iterable numbers. Hence the outcomes will be Error and 6 respectively.

9. What will be the output of the following Python function?

all(3,0,4.2)

a) True  
b) False  
c) Error  
d) 0

Answer: c  
Explanation: The function all() returns ‘True’ if any one or more of the elements of the iterable are non zero. In the above case, the values are not iterable, hence an error is thrown.

10. What will be the output of the following Python function?

min(max(False,-3,-4), 2,7)

a) 2  
b) False  
c) -3  
d) -4

Answer: b  
Explanation: The function max() is being used to find the maximum value from among -3, -4 and false. Since false amounts to the value zero, hence we are left with min(0, 2, 7) Hence the output is 0 (false).

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Built-in Functions – 2”.

1. What will be the output of the following Python functions?

chr(‘97’)

chr(97)

a)

a

Error

b)

‘a’

a

c)

Error

a

d)

Error

Error

Answer: c  
Explanation: The built-in function chr() returns the alphabet corresponding to the value given as an argument. This function accepts only integer type values. In the first function, we have passed a string. Hence the first function throws an error.

2. What will be the output of the following Python function?

complex(1+2j)

a) Error  
b) 1  
c) 2j  
d) 1+2j

Answer: d  
Explanation: The built-in function complex() returns the argument in a complex form. Hence the output of the function shown above will be 1+2j.

3. What is the output of the function complex()?  
a) 0j  
b) 0+0j  
c) 0  
d) Error

Answer: a  
Explanation: The complex function returns 0j if both of the arguments are omitted, that is, if the function is in the form of complex() or complex(0), then the output will be 0j.

4. The function divmod(a,b), where both ‘a’ and ‘b’ are integers is evaluated as:  
a) (a%b, a//b)  
b) (a//b, a%b)  
c) (a//b, a\*b)  
d) (a/b, a%b)

Answer: b  
Explanation: The function divmod(a,b) is evaluated as a//b, a%b, if both ‘a’ and ‘b’ are integers.

5. What will be the output of the following Python function?

divmod(10.5,5)

divmod(2.4,1.2)

a)

(2.00, 0.50)

(2.00, 0.00)

b)

(2, 0.5)

(2, 0)

c)

(2.0, 0.5)

(2.0, 0.0)

d)

(2, 0.5)

(2)

Answer: c  
Explanation: See python documentation for the function divmod.

6. The function complex(‘2-3j’) is valid but the function complex(‘2 – 3j’) is invalid.  
a) True  
b) False

Answer: a  
Explanation: When converting from a string, the string must not contain any blank spaces around the + or – operator. Hence the function complex(‘2 – 3j’) will result in an error.

7. What will be the output of the following Python function?

list(enumerate([2, 3]))

a) Error  
b) [(1, 2), (2, 3)]  
c) [(0, 2), (1, 3)]  
d) [(2, 3)]

Answer: c  
Explanation: The built-in function enumerate() accepts an iterable as an argument. The function shown in the above case returns containing pairs of the numbers given, starting from 0. Hence the output will be: [(0, 2), (1,3)].

8. What will be the output of the following Python functions?

x=3

eval('x^2')

a) Error  
b) 1  
c) 9  
d) 6

Answer: b  
Explanation: The function eval is use to evaluate the expression that it takes as an argument. In the above case, the eval() function is used to perform XOR operation between 3 and 2. Hence the output is 1.

9. What will be the output of the following Python functions?

float('1e-003')

float('2e+003')

a)

3.00

300

b)

0.001

2000.0

c)

0.001

200

d)

Error

2003

Answer: b  
Explanation: The output of the first function will be 0.001 and that of the second function will be 2000.0. The first function created a floating point number up to 3 decimal places and the second function adds 3 zeros after the given number.

10. Which of the following functions does not necessarily accept only iterables as arguments?  
a) enumerate()  
b) all()  
c) chr()  
d) max()

Answer: c  
Explanation: The functions enumerate(), all() and max() accept iterables as arguments whereas the function chr() throws an error on receiving an iterable as an argument. Also note that the function chr() accepts only integer values.

1. Which of the following functions accepts only integers as arguments?  
a) ord()  
b) min()  
c) chr()  
d) any()

Answer: c  
Explanation: The function chr() accepts only integers as arguments. The function ord() accepts only strings. The functions min() and max() can accept floating point as well as integer arguments.

2. Suppose there is a list such that: l=[2,3,4]. If we want to print this list in reverse order, which of the following methods should be used?  
a) reverse(l)  
b) list(reverse[(l)])  
c) reversed(l)  
d) list(reversed(l))

Answer: d  
Explanation: The built-in function reversed() can be used to reverse the elements of a list. This function accepts only an iterable as an argument. To print the output in the form of a list, we use: list(reversed(l)). The output will be: [4,3,2].

3. What will be the output of the following Python function?

float(' -12345**\n**')

(Note that the number of blank spaces before the number is 5)  
a)   -12345.0 (5 blank spaces before the number)  
b) -12345.0  
c) Error  
d) -12345.000000000…. (infinite decimal places)

Answer: b  
Explanation: The function float() will remove all the blank spaces and convert the integer to a floating point number. Hence the output will be: -12345.0.

4. What will be the output of the following Python function?

ord(65)

ord(‘A’)

a)A

65

b)Error

65

c)A

Error

d)Error

Error

Answer: b  
Explanation: The built-in function ord() is used to return the ASCII value of the alphabet passed to it as an argument. Hence the first function results in an error and the output of the second function is 65.

5. What will be the output of the following Python function?

float(‘-infinity’)

float(‘inf’)

a)–inf

inf

b)–infinity

inf

c)Error

Error

d)Error

Junk value

Answer: a  
Explanation: The output of the first function will be –inf and that of the second function will be inf.

6. Which of the following functions will not result in an error when no arguments are passed to it?  
a) min()  
b) divmod()  
c) all()  
d) float()

Answer: d  
Explanation: The built-in functions min(), max(), divmod(), ord(), any(), all() etc throw an error when no arguments are passed to them. However there are some built-in functions like float(), complex() etc which do not throw an error when no arguments are passed to them. The output of float() is 0.0.

7. What will be the output of the following Python function?

hex(15)

a) f  
b) 0xF  
c) 0Xf  
d) 0xf

Answer: d  
Explanation: The function hex() is used to convert the given argument into its hexadecimal representation, in lower case. Hence the output of the function hex(15) is 0xf.

8. Which of the following functions does not throw an error?  
a) ord()  
b) ord(‘ ‘)  
c) ord(”)  
d) ord(“”)

Answer: b  
Explanation: The function ord() accepts a character. Hence ord(), ord(”) and ord(“”) throw errors. However the function ord(‘ ‘) does not throw an error because in this case, we are actually passing a blank space as an argument. The output of ord(‘ ‘) is 32 (ASCII value corresponding to blank space).

9. What will be the output of the following Python function?

len(["hello",2, 4, 6])

a) 4  
b) 3  
c) Error  
d) 6

Answer: a  
Explanation: The function len() returns the length of the number of elements in the iterable. Therefore the output of the function shown above is 4.

10. What will be the output of the following Python function?

oct(7)

oct(‘7’)

a)Error

07

b)0o7

7

c)0o7

Error

d)07

0o7

Answer: c  
Explanation: The function oct() is used to convert its argument into octal form. This function does not accept strings. Hence the second function results in an error while the output of the first function is 0o7.

1. Which of the following is the use of function in python?  
a) Functions are reusable pieces of programs  
b) Functions don’t provide better modularity for your application  
c) you can’t also create your own functions  
d) All of the mentioned

Answer: a  
Explanation: Functions are reusable pieces of programs. They allow you to give a name to a block of statements, allowing you to run that block using the specified name anywhere in your program and any number of times.

2. Which keyword is used for function?  
a) Fun  
b) Define  
c) def  
d) Function

Answer: c  
Explanation: The def keyword is used to create, (or define) a function in python.

3. What will be the output of the following Python code?

1. def sayHello():
2. print('Hello World!')
3. sayHello()
4. sayHello()

a)

Hello World!

Hello World!

b)

'Hello World!'

'Hello World!'

c)

Hello

Hello

d) None of the mentioned

Answer: a  
Explanation: Functions are defined using the def keyword. After this keyword comes an identifier name for the function, followed by a pair of parentheses which may enclose some names of variables, and by the final colon that ends the line. Next follows the block of statements that are part of this function.

1. def sayHello():
2. print('Hello World!') # block belonging to the function
3. # End of function #
5. sayHello() # call the function
6. sayHello() # call the function again

4. What will be the output of the following Python code?

1. def printMax(a, b):
2. if a > b:
3. print(a, 'is maximum')
4. elif a == b:
5. print(a, 'is equal to', b)
6. else:
7. print(b, 'is maximum')
8. printMax(3, 4)

a) 3  
b) 4  
c) 4 is maximum  
d) None of the mentioned

Answer: c  
Explanation: Here, we define a function called printMax that uses two parameters called a and b. We find out the greater number using a simple if..else statement and then print the bigger number.

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

1. x = 50
2. def func(x):
3. print('x is', x)
4. x = 2
5. print('Changed local x to', x)
6. func(x)
7. print('x is now', x)

a)x is 50

Changed local x to 2

x is now 50

b)x is 50

Changed local x to 2

x is now 2

c)x is 50

Changed local x to 2

x is now 100

d) None of the mentioned

Answer: a  
Explanation: The first time that we print the value of the name x with the first line in the function’s body, Python uses the value of the parameter declared in the main block, above the function definition.  
Next, we assign the value 2 to x. The name x is local to our function. So, when we change the value of x in the function, the x defined in the main block remains unaffected.  
With the last print function call, we display the value of x as defined in the main block, thereby confirming that it is actually unaffected by the local assignment within the previously called function.

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

1. x = 50
2. **def** func():
3. **global** x
4. **print**('x is', x)
5. x = 2
6. **print**('Changed global x to', x)
7. func()
8. **print**('Value of x is', x)

a)

x is 50

Changed global x to 2

Value of x is 50

b)

x is 50

Changed global x to 2

Value of x is 2

c)

x is 50

Changed global x to 50

Value of x is 50

d) None of the mentioned

Answer: b  
Explanation: The global statement is used to declare that x is a global variable – hence, when we assign a value to x inside the function, that change is reflected when we use the value of x in the main block.

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

1. **def** say(message, times = 1):
2. **print**(message \* times)
3. say('Hello')
4. say('World', 5)

a)

Hello

WorldWorldWorldWorldWorld

b)

Hello

World 5

c)

Hello

World,World,World,World,World

d)

Hello

HelloHelloHelloHelloHello

Answer: a  
Explanation: For some functions, you may want to make some parameters optional and use default values in case the user does not want to provide values for them. This is done with the help of default argument values. You can specify default argument values for parameters by appending to the parameter name in the function definition the assignment operator (=) followed by the default value.  
The function named say is used to print a string as many times as specified. If we don’t supply a value, then by default, the string is printed just once. We achieve this by specifying a default argument value of 1 to the parameter times.  
In the first usage of say, we supply only the string and it prints the string once. In the second usage of say, we supply both the string and an argument 5 stating that we want to say the string message 5 times.

8. What will be the output of the following Python code?

1. def func(a, b=5, c=10):
2. print('a is', a, 'and b is', b, 'and c is', c)
4. func(3, 7)
5. func(25, c = 24)
6. func(c = 50, a = 100)

a)

a is 7 and b is 3 and c is 10

a is 25 and b is 5 and c is 24

a is 5 and b is 100 and c is 50

b)

a is 3 and b is 7 and c is 10

a is 5 and b is 25 and c is 24

a is 50 and b is 100 and c is 5

c)

a is 3 and b is 7 and c is 10

a is 25 and b is 5 and c is 24

a is 100 and b is 5 and c is 50

d) None of the mentioned

Answer: c  
Explanation: If you have some functions with many parameters and you want to specify only some of them, then you can give values for such parameters by naming them – this is called keyword arguments – we use the name (keyword) instead of the position (which we have been using all along) to specify the arguments to the function.  
The function named func has one parameter without a default argument value, followed by two parameters with default argument values.

In the first usage, func(3, 7), the parameter a gets the value 3, the parameter b gets the value 7 and c gets the default value of 10.

In the second usage func(25, c=24), the variable a gets the value of 25 due to the position of the argument. Then, the parameter c gets the value of 24 due to naming i.e. keyword arguments. The variable b gets the default value of 5.

In the third usage func(c=50, a=100), we use keyword arguments for all specified values. Notice that we are specifying the value for parameter c before that for a even though a is defined before c in the function definition.

9. What will be the output of the following Python code?

1. **def** maximum(x, y):
2. **if** x > y:
3. **return** x
4. **elif** x == y:
5. **return** 'The numbers are equal'
6. **else**:
7. **return** y
9. **print**(maximum(2, 3))

a) 2  
b) 3  
c) The numbers are equal  
d) None of the mentioned

Answer: b  
Explanation: The maximum function returns the maximum of the parameters, in this case the numbers supplied to the function. It uses a simple if..else statement to find the greater value and then returns that value.

10. Which of the following is a feature of DocString?  
a) Provide a convenient way of associating documentation with Python modules, functions, classes, and methods  
b) All functions should have a docstring  
c) Docstrings can be accessed by the \_\_doc\_\_ attribute on objects  
d) All of the mentioned

Answer: d  
Explanation: Python has a nifty feature called documentation strings, usually referred to by its shorter name docstrings. DocStrings are an important tool that you should make use of since it helps to document the program better and makes it easier to understand.

1. Which are the advantages of functions in python?  
a) Reducing duplication of code  
b) Decomposing complex problems into simpler pieces  
c) Improving clarity of the code  
d) All of the mentioned

Answer: d  
Explanation: None.

2. What are the two main types of functions?  
a) Custom function  
b) Built-in function & User defined function  
c) User function  
d) System function

Answer: b  
Explanation: Built-in functions and user defined ones. The built-in functions are part of the Python language. Examples are: dir(), len() or abs(). The user defined functions are functions created with the def keyword.

3. Where is function defined?  
a) Module  
b) Class  
c) Another function  
d) All of the mentioned

Answer: d  
Explanation: Functions can be defined inside a module, a class or another function.

4. What is called when a function is defined inside a class?  
a) Module  
b) Class  
c) Another function  
d) Method

Answer: d  
Explanation: None.

5. Which of the following is the use of id() function in python?  
a) Id returns the identity of the object  
b) Every object doesn’t have a unique id  
c) All of the mentioned  
d) None of the mentioned

Answer: a  
Explanation: Each object in Python has a unique id. The id() function returns the object’s id.

6. Which of the following refers to mathematical function?  
a) sqrt  
b) rhombus  
c) add  
d) rhombus

Answer: a  
Explanation: Functions that are always available for usage, functions that are contained within external modules, which must be imported and functions defined by a programmer with the def keyword.  
Eg: math import sqrt  
A sqrt() function is imported from the math module.

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

1. **def** cube(x):
2. **return** x \* x \* x
3. x = cube(3)
4. **print** x

a) 9  
b) 3  
c) 27  
d) 30

Answer: c  
Explanation: A function is created to do a specific task. Often there is a result from such a task. The return keyword is used to return values from a function. A function may or may not return a value. If a function does not have a return keyword, it will send a none value.

8. What will be the output of the following Python code?

1. **def** C2F(c):
2. **return** c \* 9/5 + 32
3. **print** C2F(100)
4. **print** C2F(0)

a)212

32

b)314

24

c)567

98

d) None of the mentioned

Answer: a  
Explanation: The code shown above is used to convert a temperature in degree celsius to fahrenheit.

9. What will be the output of the following Python code?

1. **def** power(x, y=2):
2. r = 1
3. **for** i **in** range(y):
4. r = r \* x
5. **return** r
6. **print** power(3)
7. **print** power(3, 3)

a)212

32

b)9

27

c)567

98

d) None of the mentioned

Answer: b  
Explanation: The arguments in Python functions may have implicit values. An implicit value is used, if no value is provided. Here we created a power function. The function has one argument with an implicit value. We can call the function with one or two arguments.

10. What will be the output of the following Python code?

1. **def** sum(\*args):
2. '''Function returns the sum
3. of all values'''
4. r = 0
5. **for** i **in** args:
6. r += i
7. **return** r
8. **print** sum.\_\_doc\_\_
9. **print** sum(1, 2, 3)
10. **print** sum(1, 2, 3, 4, 5)

a)6

15

b)6

100

c)123

12345

d) None of the mentioned

Answer: a  
Explanation: We use the \* operator to indicate, that the function will accept arbitrary number of arguments. The sum() function will return the sum of all arguments. The first string in the function body is called the function documentation string. It is used to document the function. The string must be in triple quotes.

1. Python supports the creation of anonymous functions at runtime, using a construct called \_\_\_\_\_\_\_\_\_\_  
a) lambda  
b) pi  
c) anonymous  
d) none of the mentioned

Answer: a  
Explanation: Python supports the creation of anonymous functions (i.e. functions that are not bound to a name) at runtime, using a construct called lambda. Lambda functions are restricted to a single expression. They can be used wherever normal functions can be used.

2. What will be the output of the following Python code?

1. y = 6
2. z = **lambda** x: x \* y
3. **print** z(8)

a) 48  
b) 14  
c) 64  
d) None of the mentioned

Answer: a  
Explanation: The lambda keyword creates an anonymous function. The x is a parameter, that is passed to the lambda function. The parameter is followed by a colon character. The code next to the colon is the expression that is executed, when the lambda function is called. The lambda function is assigned to the z variable.  
The lambda function is executed. The number 8 is passed to the anonymous function and it returns 48 as the result. Note that z is not a name for this function. It is only a variable to which the anonymous function was assigned.

advertisement

3. What will be the output of the following Python code?

1. lamb = **lambda** x: x \*\* 3
2. **print**(lamb(5))

a) 15  
b) 555  
c) 125  
d) None of the mentioned

Answer: c  
Explanation: None.

4. Does Lambda contains return statements?  
a) True  
b) False

Answer: b  
Explanation: lambda definition does not include a return statement. it always contains an expression which is returned. Also note that we can put a lambda definition anywhere a function is expected. We don’t have to assign it to a variable at all.

5. Lambda is a statement.  
a) True  
b) False

Answer: b  
Explanation: lambda is an anonymous function in Python. Hence this statement is false.

6. Lambda contains block of statements.  
a) True  
b) False

Answer: b  
Explanation: None.

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

1. **def** f(x, y, z): **return** x + y + z
2. f(2, 30, 400)

a) 432  
b) 24000  
c) 430  
d) No output

Answer: a  
Explanation: None.

8. What will be the output of the following Python code?

1. **def** writer():
2. title = 'Sir'
3. name = (**lambda** x:title + ' ' + x)
4. **return** name
6. who = writer()
7. **print**(who('Arthur'))

a) Arthur Sir  
b) Sir Arthur  
c) Arthur  
d) None of the mentioned

Answer: b  
Explanation: None.

9. What will be the output of the following Python code?

1. L = [**lambda** x: x \*\* 2,
2. **lambda** x: x \*\* 3,
3. **lambda** x: x \*\* 4]
5. **for** f **in** L:
6. **print**(f(3))

a)27

81

343

b)6

9

12

c)9

27

81

d) None of the mentioned

Answer: c  
Explanation: None.

10. What will be the output of the following Python code?

1. min = (**lambda** x, y: x **if** x < y **else** y)
2. min(101\*99, 102\*98)

a) 9997  
b) 9999  
c) 9996  
d) None of the mentioned

Answer: c  
Explanation: None.

1. What is a variable defined outside a function referred to as?  
a) A static variable  
b) A global variable  
c) A local variable  
d) An automatic variable

Answer: b  
Explanation: The value of a variable defined outside all function definitions is referred to as a global variable and can be used by multiple functions of the program.

2. What is a variable defined inside a function referred to as?  
a) A global variable  
b) A volatile variable  
c) A local variable  
d) An automatic variable

Answer: c  
Explanation: The variable inside a function is called as local variable and the variable definition is confined only to that function.

3. What will be the output of the following Python code?

i=0

**def** change(i):

i=i+1

**return** i

change(1)

**print**(i)

a) 1  
b) Nothing is displayed  
c) 0  
d) An exception is thrown

Answer: c  
Explanation: Any change made in to an immutable data type in a function isn’t reflected outside the function.

4. What will be the output of the following Python code?

**def** a(b):

b = b + [5]

c = [1, 2, 3, 4]

a(c)

**print**(len(c))

a) 4  
b) 5  
c) 1  
d) An exception is thrown

Answer: a  
Explanation: Since a list is mutable, any change made in the list in the function is reflected outside the function.

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

a=10

b=20

**def** change():

**global** b

a=45

b=56

change()

**print**(a)

**print**(b)

a)10

56

b)45

56

c)10

20

d) Syntax Error

Answer: a  
Explanation: The statement “global b” allows the global value of b to be accessed and changed. Whereas the variable a is local and hence the change isn’t reflected outside the function.

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

**def** change(i = 1, j = 2):

i = i + j

j = j + 1

**print**(i, j)

change(j = 1, i = 2)

a) An exception is thrown because of conflicting values  
b) 1 2  
c) 3 3  
d) 3 2

Answer: d  
Explanation: The values given during function call is taken into consideration, that is, i=2 and j=1.

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

**def** change(one, \*two):

**print**(type(two))

change(1,2,3,4)

a) Integer  
b) Tuple  
c) Dictionary  
d) An exception is thrown

Answer: b  
Explanation: The parameter two is a variable parameter and consists of (2,3,4). Hence the data type is tuple.

8. If a function doesn’t have a return statement, which of the following does the function return?  
a) int  
b) null  
c) None  
d) An exception is thrown without the return statement

Answer: c  
Explanation: A function can exist without a return statement and returns None if the function doesn’t have a return statement.

9. What will be the output of the following Python code?

**def** display(b, n):

**while** n > 0:

**print**(b,end="")

n=n-1

display('z',3)

a) zzz  
b) zz  
c) An exception is executed  
d) Infinite loop

Answer: a  
Explanation: The loop runs three times and ‘z’ is printed each time.

10. What will be the output of the following Python code?

**def** find(a, \*\*b):

**print**(type(b))

find('letters',A='1',B='2')

a) String  
b) Tuple  
c) Dictionary  
d) An exception is thrown

Answer: c  
Explanation: b combines the remaining parameters into a dictionary.