Create a function called mult that has two parameters, the first is required and should be an integer, the second is an optional parameter that can either be a number or a string but whose default is 6. The function should return the first parameter multiplied by the second.

def mult (take, number = 6):

return take \* number

t = 15

print (mult (15))

The following function, greeting, does not work. Please fix the code so that it runs without error. This only requires one change in the definition of the function.

def greeting (name, greeting="Hello ", excl="!"):

return greeting + name + excl

print(greeting("Bob"))

print (greeting (""))

print (greeting ("Bob", excl="!!!"))

Below is a function, sum, that does not work. Change the function definition so the code works. The function should still have a required parameter, intx, and an optional parameter, intz with a defualt value of 5.

def sum (intx, intz=5):

return intz + intx

x = 5

print(sum(x))

Write a function, test, that takes in three parameters: a required integer, an optional boolean whose default value is True, and an optional dictionary, called dict1, whose default value is {2:3, 4:5, 6:8}. If the boolean parameter is True, the function should test to see if the integer is a key in the dictionary. The value of that key should then be returned. If the boolean parameter is False, return the boolean value “False”.

def test (x, b = True, dict1 = {2:3, 4:5, 6:8}):

if b == True:

if x in dict1:

return dict1[x]

else:

return False

n = 0

print(test(n))

Write a function called checkingIfIn that takes three parameters. The first is a required parameter, which should be a string. The second is an optional parameter called direction with a default value of True. The third is an optional parameter called d that has a default value of {'apple': 2, 'pear': 1, 'fruit': 19, 'orange': 5, 'banana': 3, 'grapes': 2, 'watermelon': 7}. Write the function checkingIfIn so that when the second parameter is True, it checks to see if the first parameter is a key in the third parameter; if it is, return True, otherwise return False.

But if the second paramter is False, then the function should check to see if the first parameter is not a key of the third. If it’s not, the function should return True in this case, and if it is, it should return False.

def checkingIfIn (string, direction = True, d = {'apple': 2, 'pear': 1, 'fruit': 19, 'orange': 5, 'banana': 3, 'grapes': 2, 'watermelon': 7}):

if direction == True:

if string in d:

return True

else:

return False

else:

if string not in d:

return True

else:

return False

s = "pear"

print(checkingIfIn(s))

We have provided the function checkingIfIn such that if the first input parameter is in the third, dictionary, input parameter, then the function returns that value, and otherwise, it returns False. Follow the instructions in the active code window for specific variable assignmemts.

def checkingIfIn (a, direction = True, d = {'apple': 2, 'pear': 1, 'fruit': 19, 'orange': 5, 'banana': 3, 'grapes': 2, 'watermelon': 7}):

if direction == True:

if a in d:

return d[a]

else:

return False

else:

if a not in d:

return True

else:

return d[a]

# Call the function so that it returns False and assign that function call to the variable c\_false

c\_false = checkingIfIn("Lima")

# Call the fucntion so that it returns True and assign it to the variable c\_true

c\_true = checkingIfIn ('apple', False, {'peas': 1, 'fruit': 19, 'orange': 5, 'banana': 3})

# Call the function so that the value of fruit is assigned to the variable fruit\_ans

fruit\_ans = checkingIfIn('fruit')

# Call the function using the first and third parameter so that the value 8 is assigned to the variable param\_check

param\_check = checkingIfIn ('apple', False, {'apple': 8, 'peas': 1, 'fruit': 19, 'orange': 5, 'banana': 3})