**Part 1**

1. Answer: False

Since the return function will break the iterative loop after checking the condition of the if statement on the first character of the string

1. Answer: False

Since the ‘if condition’ has a ‘c’ and not c (which is the correct syntax for a for loop) it will always return True, since ‘c’ is lowercase. If we replace ‘c’ with capital ‘C’, it will return false. Basically, the if condition is only checking for the character ‘c’ and not traversing all the characters of the string. This is a clear case of syntax error.

1. Answer: False

The for loop keeps updating the variable ‘flag’ through the loop and will return the Boolean condition of just the last character

1. Answer: True

Since the variable ‘flag’ is kept outside the for loop and the flag statement inside the ‘for’ loop has an ‘or’ condition, the function will return the correct checks for lowercase letters every time.

1. Answer: False

Like in 1 above, in this case also, the ‘for’ loop will be exited at the first instance of an uppercase character without iterating through the entire string

**Part 2**

rotate\_string = input("Enter the string for rotation: ")

rotate\_number = int(input("Enter the number for rotation: "))

new\_word = []

def rotate\_word (rotate\_number, rotate\_string):

for letter in rotate\_string:

if letter. islower():

count\_start = 97

count\_end = 122

new\_letter\_number = (ord(letter) + (rotate\_number))

if new\_letter\_number >= 122:

to\_scale = (new\_letter\_number - count\_end) % 25

if to\_scale > 0:

new\_letter\_number = (count\_start + to\_scale - 1)

elif to\_scale == 0:

new\_letter\_number = ord(letter)

elif new\_letter\_number <= 97:

to\_scale = (count\_start - new\_letter\_number) % 25

if to\_scale > 0:

new\_letter\_number = (count\_end - to\_scale + 1)

elif to\_scale == 0:

new\_letter\_number = ord(letter)

else:

new\_letter\_number

else:

count\_start = 64

count\_end = 90

new\_letter\_number = (ord(letter) + (rotate\_number))

if new\_letter\_number >= 90:

to\_scale = (new\_letter\_number - count\_end) % 25

if to\_scale > 0:

new\_letter\_number = (count\_start + to\_scale - 1)

elif to\_scale == 0:

new\_letter\_number = ord(letter)

elif to\_scale < 0:

new\_letter\_number = (count\_end + to\_scale)

elif new\_letter\_number <= 64:

to\_scale = (count\_start - abs(new\_letter\_number)) % 25

if to\_scale > 0:

new\_letter\_number = (count\_end - to\_scale + 1)

elif to\_scale == 0:

new\_letter\_number = ord(letter)

else:

new\_letter\_number

new\_letter = chr(new\_letter\_number)

new\_word.append(new\_letter)

new\_word\_complete = ''.join(new\_word)

print(new\_word\_complete)

rotate\_word(rotate\_number,rotate\_string)