Question 3

Part 1

a) **By decrypting the code, initially I try to fix the error in the code. though it reveals the original code with many errors. The code presents to be ROT13-encrypted. In it each letter is moved 13 positions in the alphabet.**

global Variable 100

a1_dict ('key1': 'value1', 'key2': 'Value2', 'key3': 'Value3']

def process_numbers(): global global Variable local Variable 5 numbers [1, 2, 3, 4, 5]

while local_variable > 0: if local_variable % 2 == 0: numbers.remove(local_variable) local variable 1

return numbers

my_set (1, 2, 3, 4, 5, 5, 4, 3, 2, 1) result process_numbers(numbers=my_set)

def modify_dict(): local variable 10 a1_dict['key4'] local_variable

modify_dict(5)

def update_global(): global global_variable global_variable += 10

for i in range(5):

print(i) I1

if my_set is not None and a1_dict['key4'] == 10: print("Condition met!")

if 5 not in my_dict: print("5 not found in the dictionary!")

print(global Variable) print(my_dict)

print(my_set)

b) Now I tried to write the accurate version of the code with the explanation of using the comments (#) below:

```
global_variable = 100  # Fix: Python variable names should be lowercase with underscores

a1_dict = {'key1': 'value1', 'key2': 'Value2', 'key3': 'Value3'}  # Fix: Correct dictionary definition with proper syntax

def process_numbers():

    global global_variable  # Fix: Accessing global variable requires proper declaration with the 'global' keyword

    local_variable = 5  # Fix: Assign value to 'local_variable'

    numbers = [1, 2, 3, 4, 5]  # Fix: 'numbers' list syntax corrected


    while local_variable > 0:  # Fix: Use 'local_variable' correctly

        if local_variable % 2 == 0:  # Fix: Correct modulo condition for even numbers

            numbers.remove(local_variable)  # Fix: Use 'remove' method properly

        local_variable -= 1  # Fix: Decrement 'local_variable' to avoid infinite loop


    return numbers  # Return the modified list


my_set = {1, 2, 3, 4, 5, 5, 4, 3, 2, 1}  # Fix: Sets should use curly braces to eliminate duplicate elements

result = process_numbers()  # Fix: 'process_numbers' does not take parameters, remove argument


def modify_dict():

    local_variable = 10  # Fix: Declare 'local_variable' inside the function

    a1_dict['key4'] = local_variable  # Fix: Properly assign 'local_variable' to a new dictionary key


modify_dict()  # Fix: Call the function without any arguments


def update_global():
```

```
    global global_variable  # Fix: Declare the use of 'global_variable'
    global_variable += 10  # Fix: Correct variable update

for i in range(5):  # Fix: Python loop syntax for range
    print(i)  # Fix: Proper 'print' statement

if my_set is not None and a1_dict['key4'] == 10:  # Fix: Check if the set is not None and the dictionary value equals 10
    print("Condition met!")

if 5 not in a1_dict:  # Fix: Check if key 5 is not in the dictionary
    print("5 not found in the dictionary!")

print(global_variable)  # Fix: Print the global variable value
print(a1_dict)  # Fix: Print the dictionary
print(my_set)  # Fix: Print the set
```