Question 3

Answer:

From the encrypt(text,key) function, we can find that ord(char) will turn char into ASCII number and shift the char backward by the number of key. For example, if k=3, char ‘A’ will shift backward in ASCII and become “D”. If we shift the char in the reverse direction by key number which means shift char forward, we can get the decryption function which is shown in the following and in the decrypt.py. The importance in decrypt(text, key) is to “shifted = ord(char) - key” and we can shift char backward. Meanwhile, we use the code in image 1 to get total number is 13. Thus, the key is 13 as well. Now, we have the decrypt function.

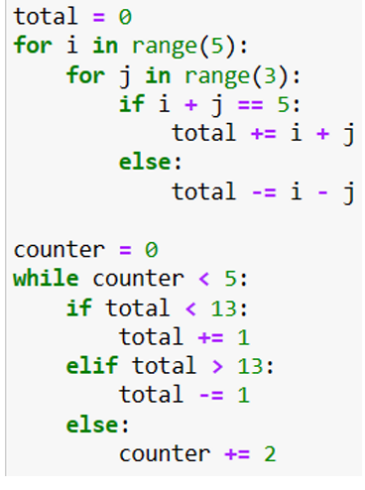
Next, we can load encrypted code which is stored in the encrpted\_code.txt into decrypt() function by with open() method and file.read() method. This code is also shown in the following part and the decrypt.py file. By running decrypt.py and decrypt(text,key) function, we can get decrypted code which is stored in the decrypted\_code.py file.

The errors in the decrypted\_code.py is shown in the following and in the decrypted\_code\_modified.py file.

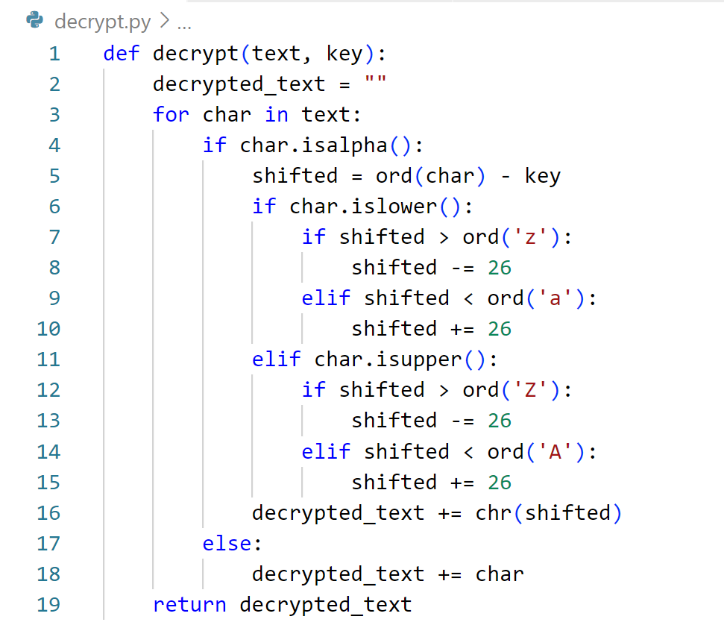
1. Accord to result=process\_numbers(numbers=my\_set), the process\_numbers() need a parameter. Therefore, we add numbers into process\_numbers(). Def process\_numbers(numbers) function is used to remove even elements from my\_set.
2. The parameter 5 in modify\_dict(5) is meaningless because def modify\_dict() doesn’t use external parameter.
3. def update\_global() is defined but isn’t called in the original code. This function is meaningless unless we call it. Thus, we add update\_global() to call def update\_global().

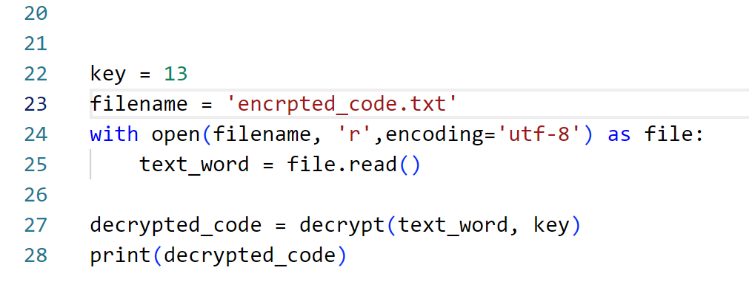
After modification on decrypted\_code.py, we can get decrypted\_code\_modified.py. By running this code, we can get the result shown in the following.

Image 1:

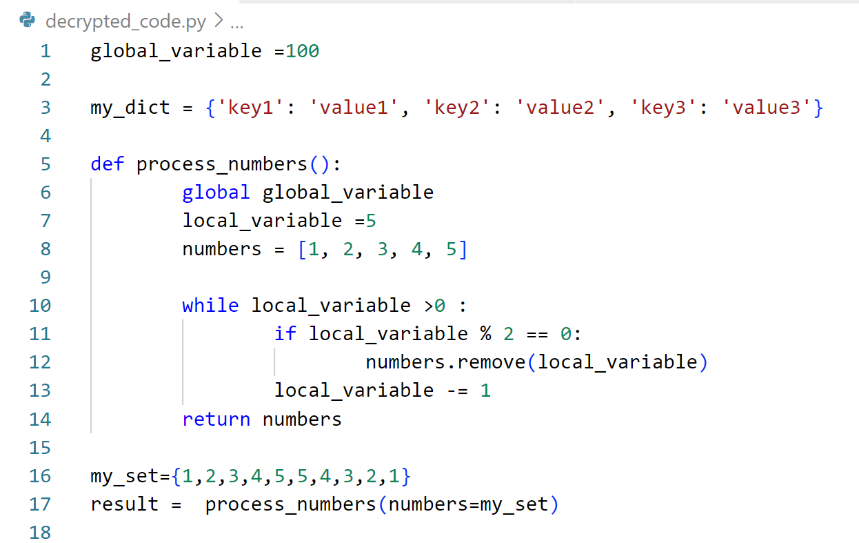


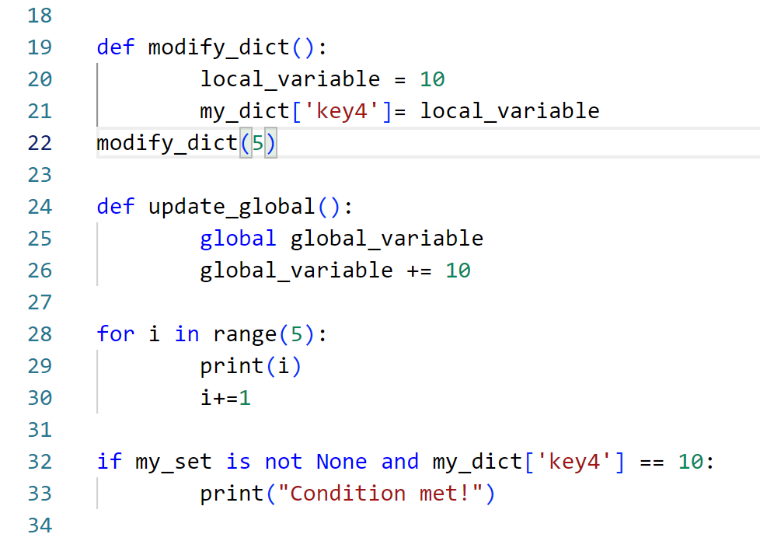
decrypt.py:

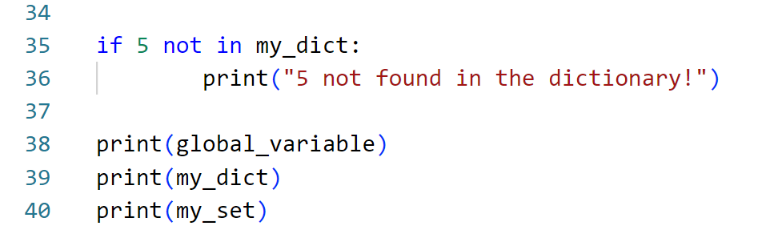




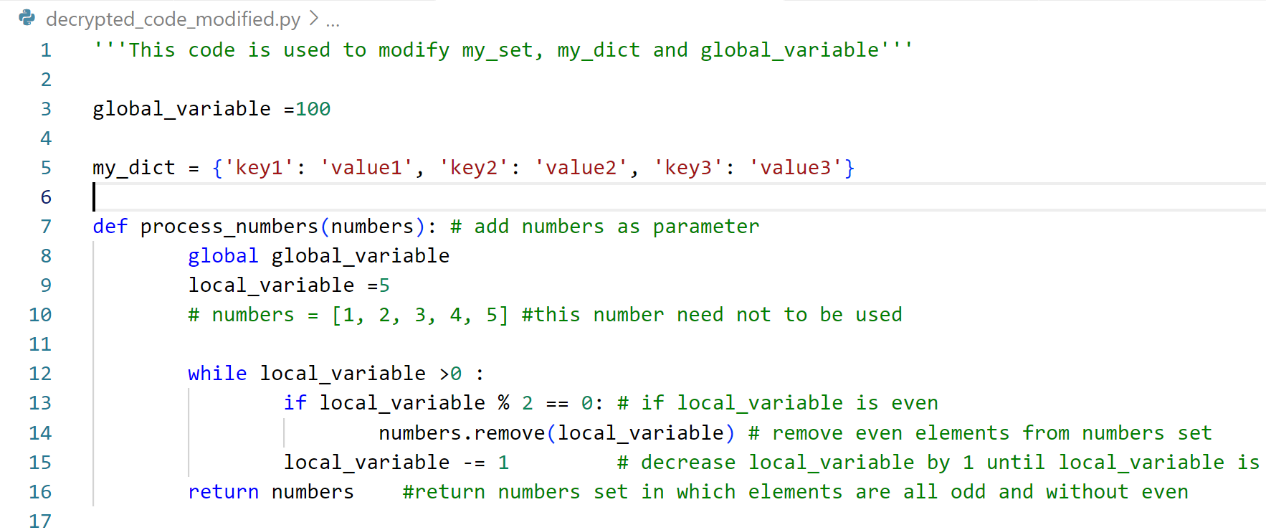
decrypted\_code.py:

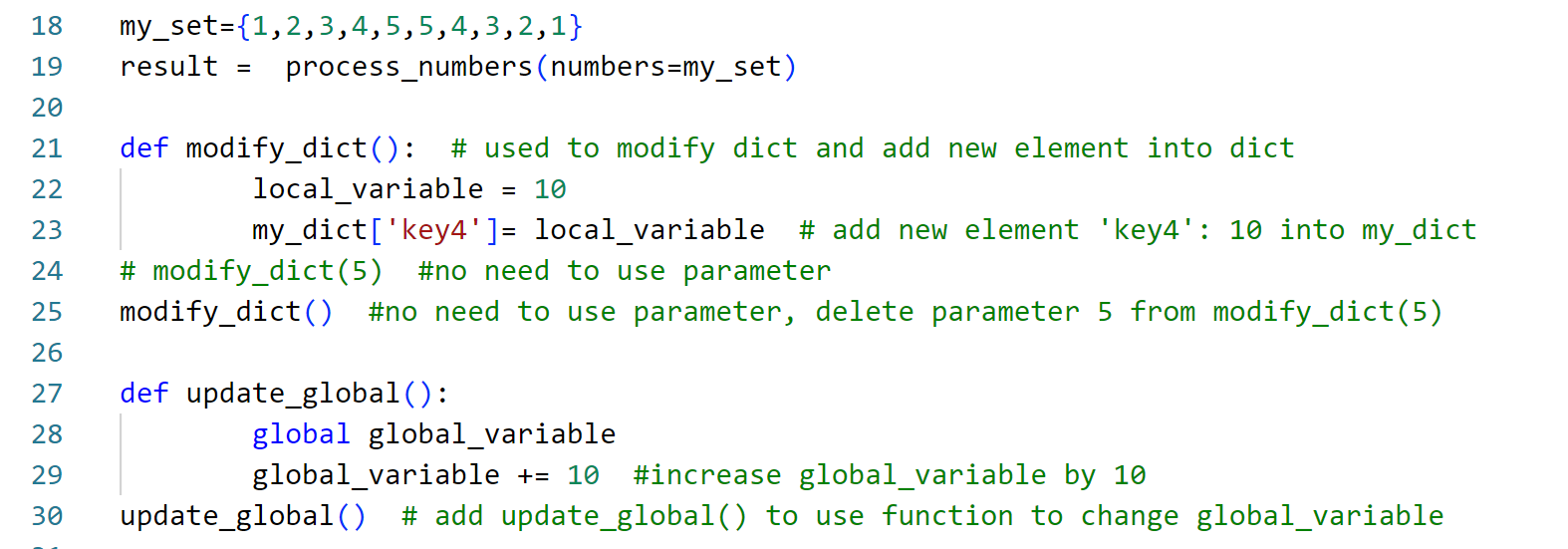


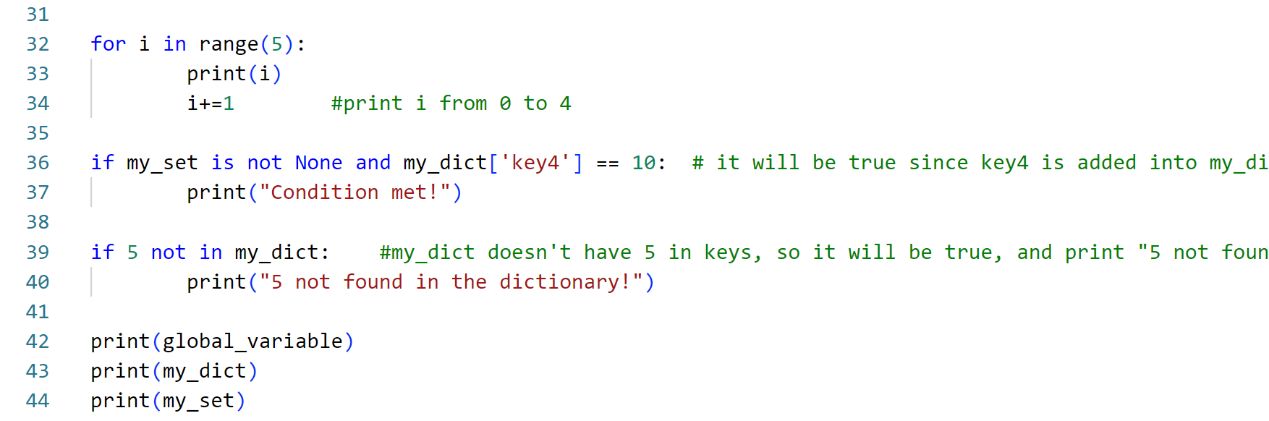




Decrypted\_code\_modified.py







The result is:

0

1

2

3

4

Condition met!

5 not found in the dictionary!

110

{'key1': 'value1', 'key2': 'value2', 'key3': 'value3', 'key4': 10}

{1, 3, 5}