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.

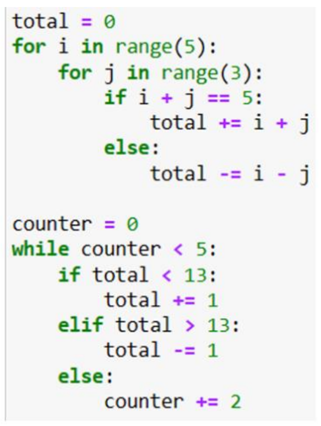
Next, we can load encrypted code which is stored in the encrpted_code.txt into decrypt() function by with open() method and file.read() method. This code is also shown in the following part and the decrypt.py file. By running decrypt.py and decrypt(text,key) function, we can get decrypted code which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

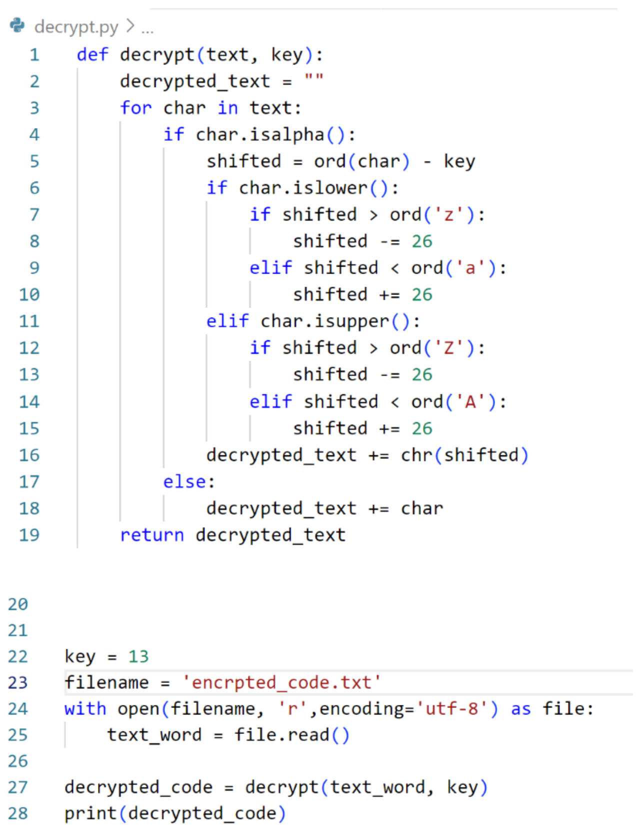
After modification on decrypted_code.py, we can get decrypted_code_modified.py. By running this code, we can get the result shown in the following.

Image 1:



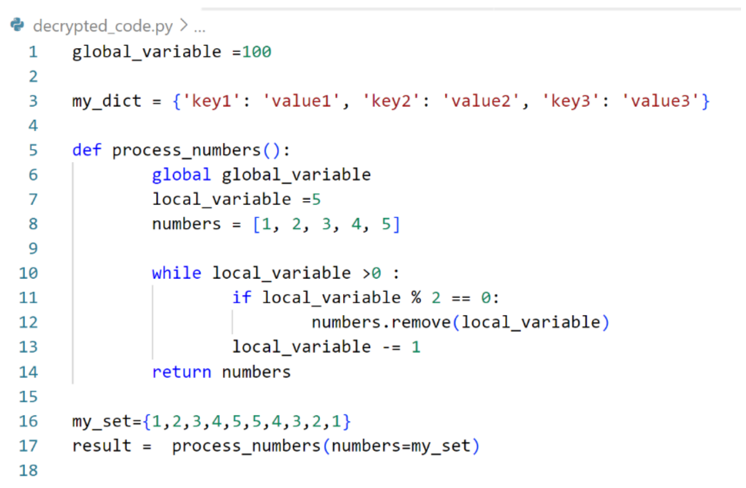
```
total = 0
for i in range(5):
    for j in range(3):
        if i + j == 5:
            total += i + j
        else:
            total -= i - j

counter = 0
while counter < 5:
    if total < 13:
        total += 1
    elif total > 13:
        total -= 1
    else:
        counter += 2
```

decrypt.py:

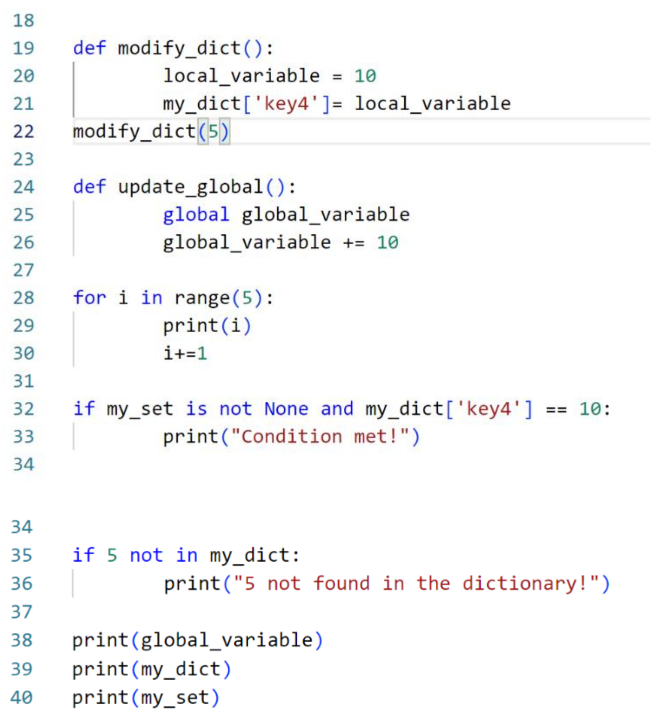


```python
def decrypt(text, key):
    decrypted_text = ""
    for char in text:
        if char.isalpha():
            shifted = ord(char) - key
            if char.islower():
                if shifted > ord('z'):
                    shifted -= 26
                elif shifted < ord('a'):
                    shifted += 26
            elif char.isupper():
                if shifted > ord('Z'):
                    shifted -= 26
                elif shifted < ord('A'):
                    shifted += 26
            decrypted_text += chr(shifted)
        else:
            decrypted_text += char
    return decrypted_text



key = 13
filename = 'encrpted_code.txt'
with open(filename, 'r',encoding='utf-8') as file:
    text_word = file.read()

decrypted_code = decrypt(text_word, key)
print(decrypted_code)
```

decrypted_code.py:

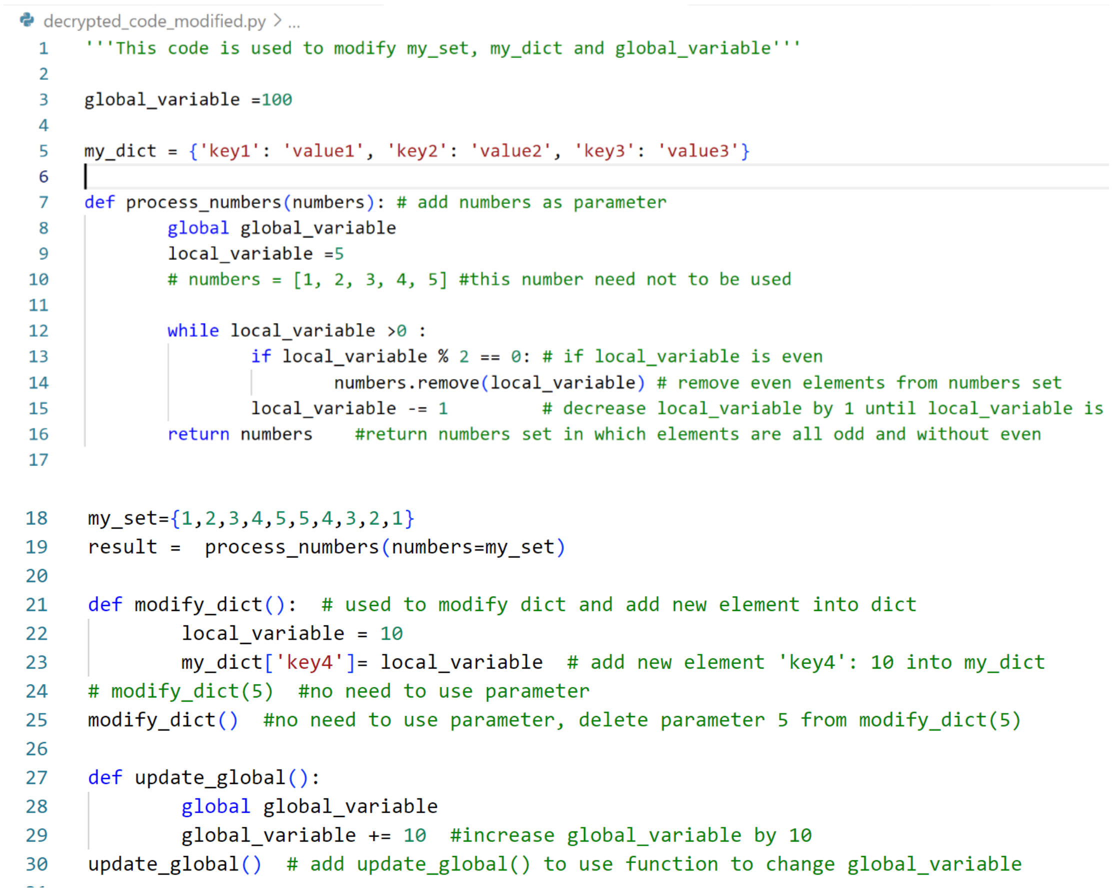


```python
global_variable =100

my_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)

```



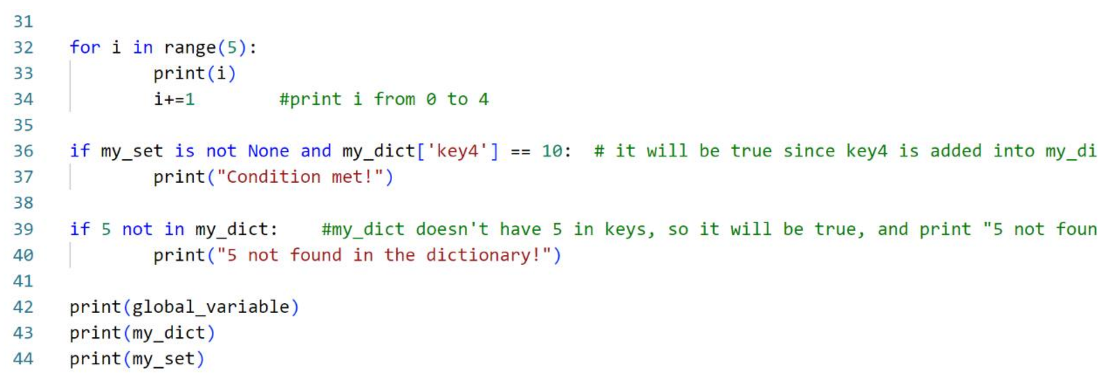
```python

def modify_dict():
        local_variable = 10
        my_dict['key4']= local_variable
modify_dict(5)

def update_global():
        global global_variable
        global_variable += 10

for i in range(5):
        print(i)
        i+=1

if my_set is not None and my_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)
```

Decrypted_code_modified.py



decrypted_code_modified.py > ...

```python
'''This code is used to modify my_set, my_dict and global_variable'''

global_variable =100

my_dict = {'key1': 'value1', 'key2': 'value2', 'key3': 'value3'}

def process_numbers(numbers): # add numbers as parameter
        global global_variable
        local_variable =5
        # numbers = [1, 2, 3, 4, 5] #this number need not to be used

        while local_variable >0 :
                if local_variable % 2 == 0: # if local_variable is even
                        numbers.remove(local_variable) # remove even elements from numbers set
                local_variable -= 1         # decrease local_variable by 1 until local_variable is
        return numbers    #return numbers set in which elements are all odd and without even


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

def modify_dict():  # used to modify dict and add new element into dict
        local_variable = 10
        my_dict['key4']= local_variable  # add new element 'key4': 10 into my_dict
# modify_dict(5)  #no need to use parameter
modify_dict()  #no need to use parameter, delete parameter 5 from modify_dict(5)

def update_global():
        global global_variable
        global_variable += 10  #increase global_variable by 10
update_global()  # add update_global() to use function to change global_variable
```



```python

for i in range(5):
        print(i)
        i+=1        #print i from 0 to 4

if my_set is not None and my_dict['key4'] == 10:  # it will be true since key4 is added into my_di
        print("Condition met!")

if 5 not in my_dict:    #my_dict doesn't have 5 in keys, so it will be true, and print "5 not foun
        print("5 not found in the dictionary!")

print(global_variable)
print(my_dict)
print(my_set)
```

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}
```