d. m4.py uses the code of list1.py, set2.py and dict3.py

# m4.py

import list1

import set2

import dict3

def demo\_list\_operations():

print("List Operations Demo:")

list1.append1(10)

list1.append1(20)

list1.extend1([30, 40, 50])

print("Current List:", list1.get\_list())

list1.pop()

print("List after pop:", list1.get\_list())

list1.remove1(20)

print("List after removing 20:", list1.get\_list())

print("-" \* 30)

def demo\_set\_operations():

print("Set Operations Demo:")

set2.adds2(100)

set2.adds2(200)

set2.adds2(300)

print("Current Set:", set2.get\_set())

set2.remove2(200)

print("Set after removing 200:", set2.get\_set())

sample\_set = {1, 2, 3}

print("Length of sample set {1,2,3}:", set2.slen2(sample\_set))

print("-" \* 30)

def demo\_dict\_operations():

print("Dictionary Operations Demo:")

dict3.add3('name', 'Alice')

dict3.add3('age', 25)

print("Current Dictionary:", dict3.get\_dict())

print("Keys:", dict3.keys3())

print("Values:", dict3.values3())

sample\_dict = {'x': 1, 'y': 2, 'z': 3}

print("Length of sample dict {'x':1, 'y':2, 'z':3}:", dict3.len3(sample\_dict))

print("-" \* 30)

if \_\_name\_\_ == "\_\_main\_\_":

demo\_list\_operations()

demo\_set\_operations()

demo\_dict\_operations()