CMPSC 462 – Assignment-1 (30 points)
Lists and Dictionaries

Due date: 8/30/2022

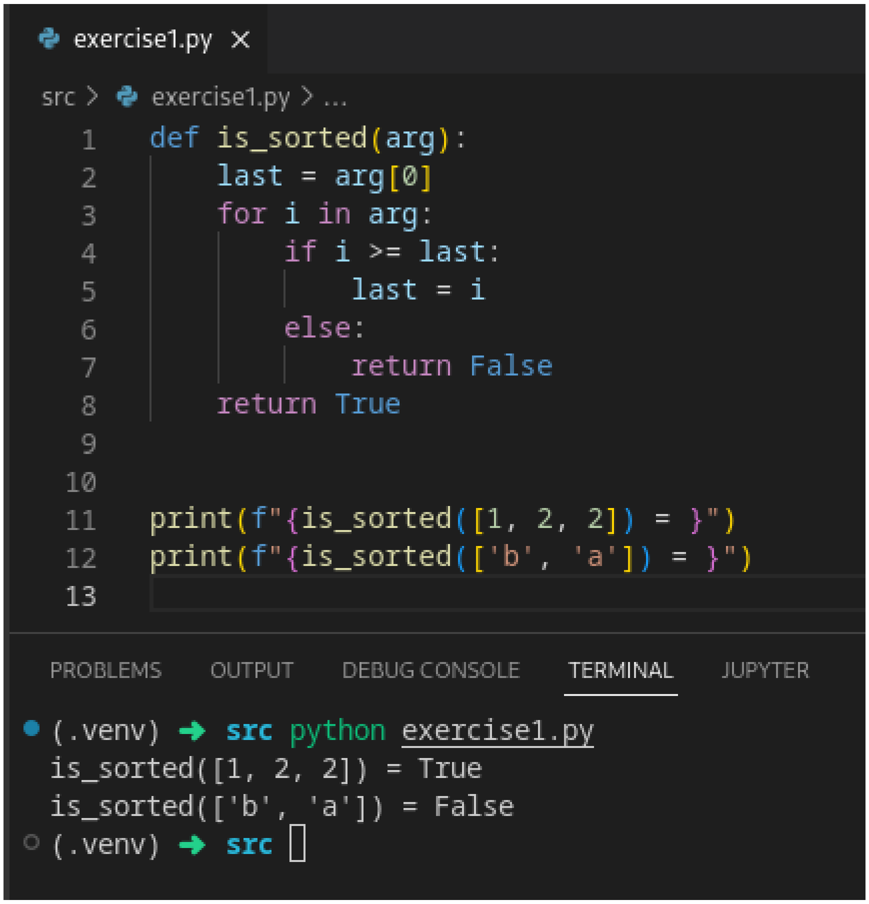
Note: attach screenshots of your program and results under each programming exercises. Please make sure that the screenshot is readable. Don't attach a very small screenshot image.

Exercise:(1-5) - 6 points each

Exercise-1:

Write a function called is_sorted (without using inbuilt sort function) that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise. You can assume (as a precondition) that the elements of the list can be compared with the relational operators <, >, etc.

For example, is_sorted([1,2,2]) should return True and is_sorted(['b','a']) should return False.



```python
def is_sorted(arg):
    last = arg[0]
    for i in arg:
        if i >= last:
            last = i
        else:
            return False
    return True


print(f"{is_sorted([1, 2, 2]) = }")
print(f"{is_sorted(['b', 'a']) = }")

```

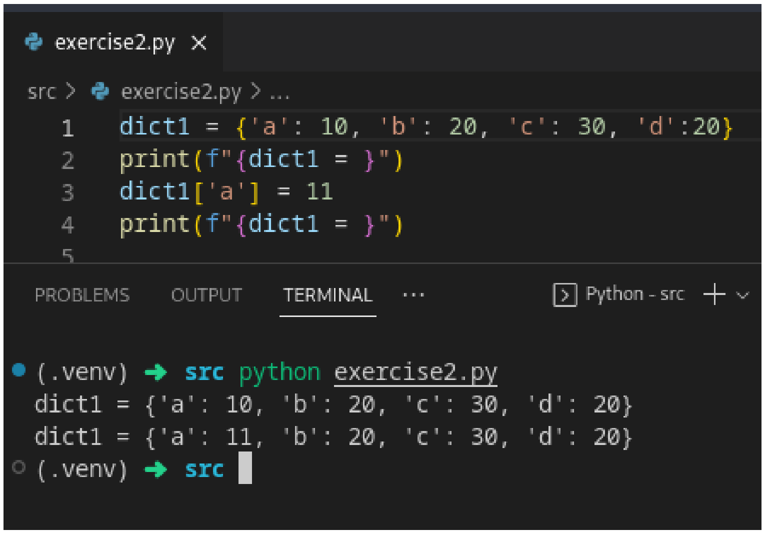
```
● (.venv) ➜ src python exercise1.py
is_sorted([1, 2, 2]) = True
is_sorted(['b', 'a']) = False
○ (.venv) ➜ src
```

Exercise-2:

What command you would use to do the following for this dictionary:

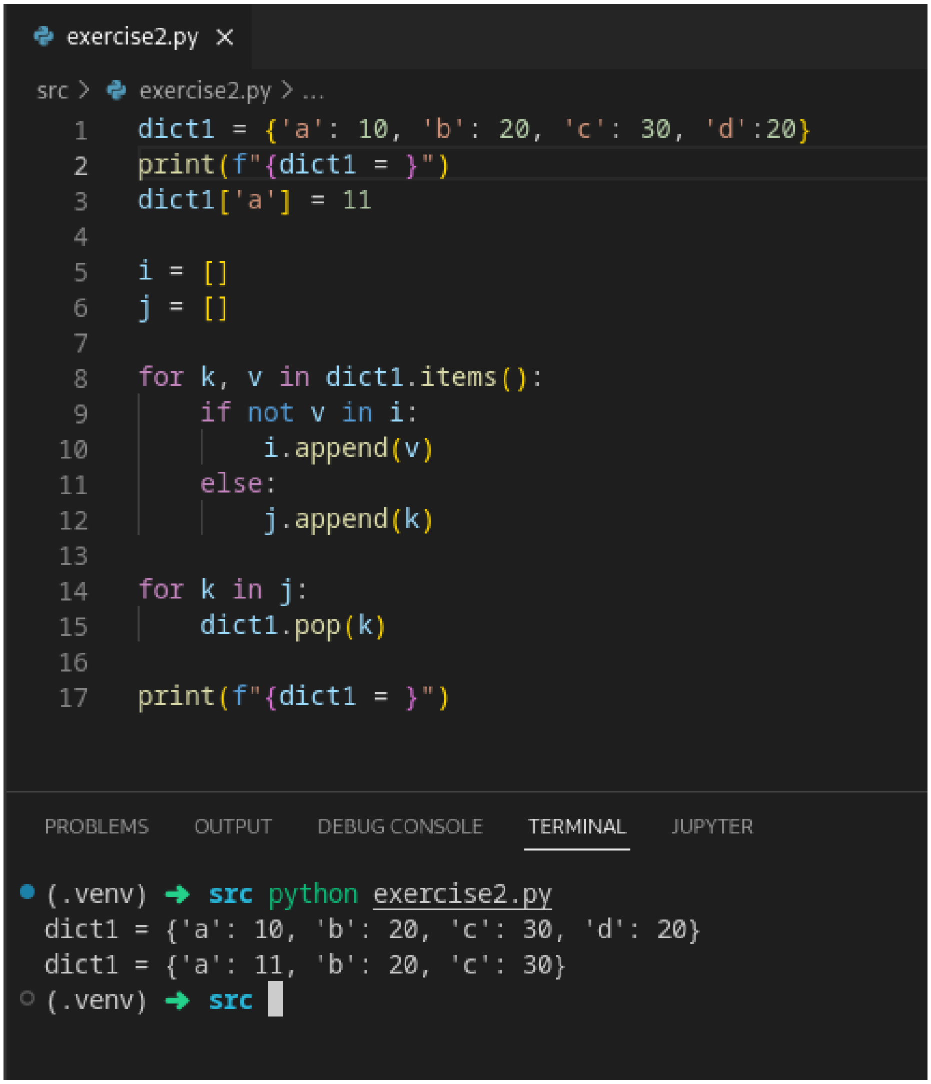
dict1 = {'a': 10, 'b': 20, 'c': 30, ‘d’:20}

1. Update an entry in dict1
2. Show how to remove the duplicate values from dict1



```python
dict1 = {'a': 10, 'b': 20, 'c': 30, 'd':20}
print(f"{dict1 = }")
dict1['a'] = 11
print(f"{dict1 = }")

```

```
(.venv) ➜ src python exercise2.py
dict1 = {'a': 10, 'b': 20, 'c': 30, 'd': 20}
dict1 = {'a': 11, 'b': 20, 'c': 30, 'd': 20}
(.venv) ➜ src 
```

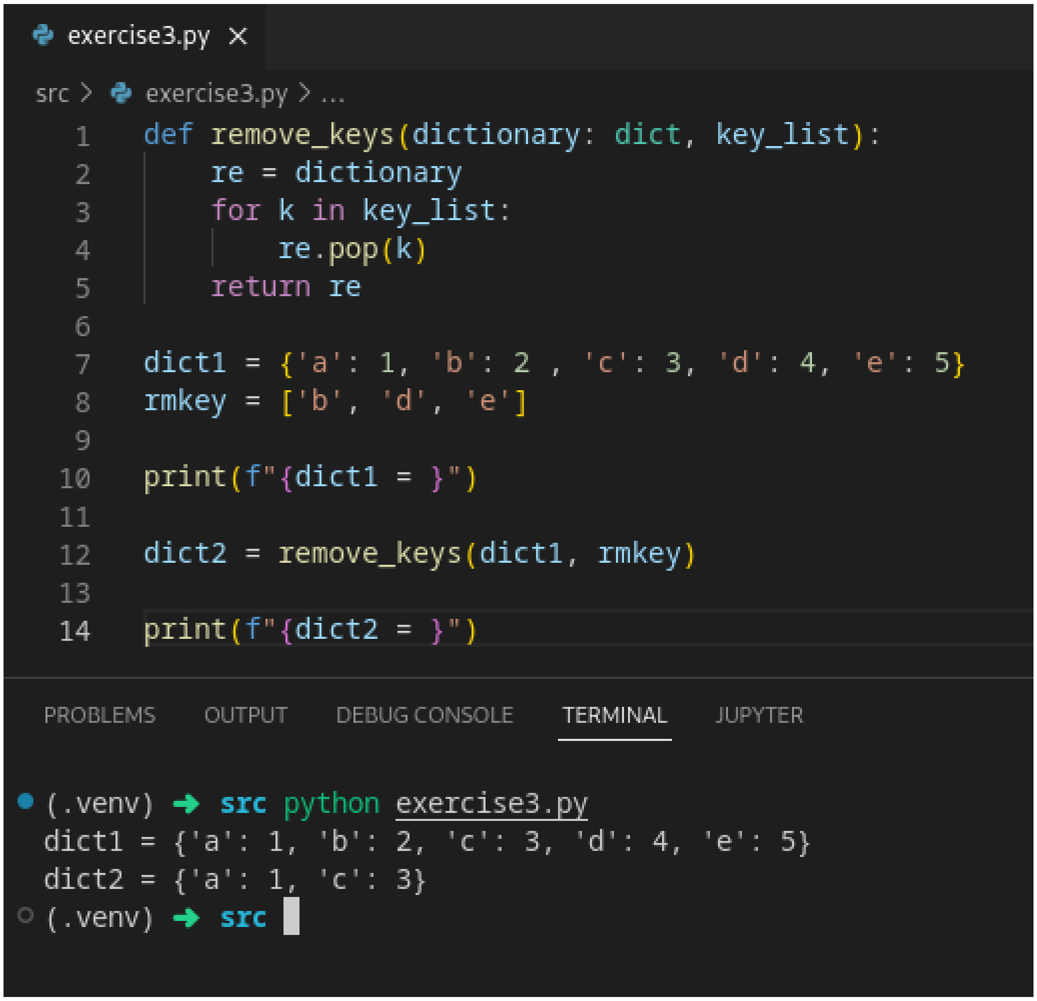


```python
dict1 = {'a': 10, 'b': 20, 'c': 30, 'd':20}
print(f"{dict1 = }")
dict1['a'] = 11

i = []
j = []

for k, v in dict1.items():
    if not v in i:
        i.append(v)
    else:
        j.append(k)

for k in j:
    dict1.pop(k)

print(f"{dict1 = }")
```

```
(.venv) ➜ src python exercise2.py
dict1 = {'a': 10, 'b': 20, 'c': 30, 'd': 20}
dict1 = {'a': 11, 'b': 20, 'c': 30}
(.venv) ➜ src 
```

Exercise-3:
Write a function called remove keys(mydict, keylist) that accepts two parameters: a dictionary called mydict and a list called keylist. remove keys(mydict, keylist) should remove all the keys contained in keylist from mydict and return the dictionary:

```
d = { "key1" : "value1", "key2" : "value2", "key3" : "value3", "key4" :
"value4" }
keys = ["key1", "key3", "key5"]
```



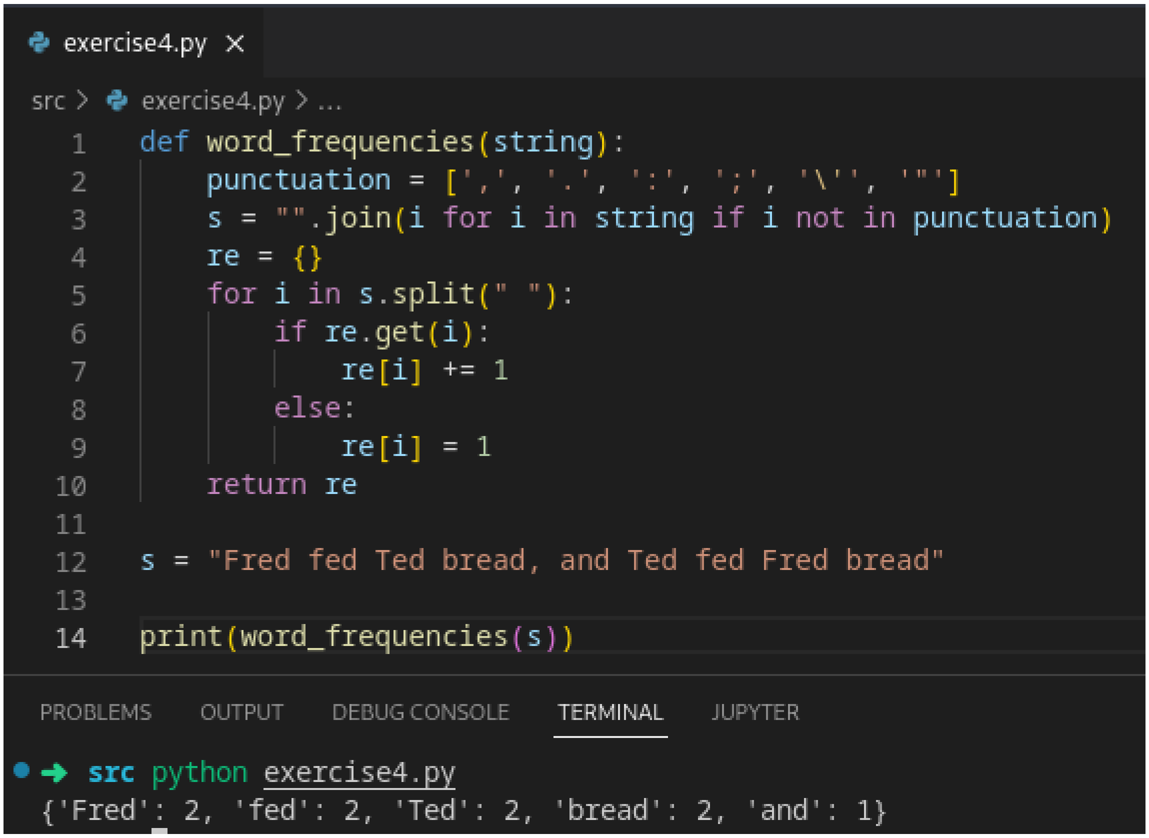
```python
def remove_keys(dictionary: dict, key_list):
    re = dictionary
    for k in key_list:
        re.pop(k)
    return re

dict1 = {'a': 1, 'b': 2 , 'c': 3, 'd': 4, 'e': 5}
rmkey = ['b', 'd', 'e']

print(f"{dict1 = }")

dict2 = remove_keys(dict1, rmkey)

print(f"{dict2 = }")
```

```
● (.venv) ➜ src python exercise3.py
  dict1 = {'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}
  dict2 = {'a': 1, 'c': 3}
○ (.venv) ➜ src
```

Exercise-4:
Write a function called word frequencies(mylist) that accepts a strings of words and returns a dictionary where the keys are the words from the string of words and the values are the number of times that word appears in mylist:

S = “Fred fed Ted bread, and Ted fed Fred bread”

```
word_freq = { ’Fred’:2, ’fed’:2, ’Ted’:2, ’bread’:2, ’and’:1}
```



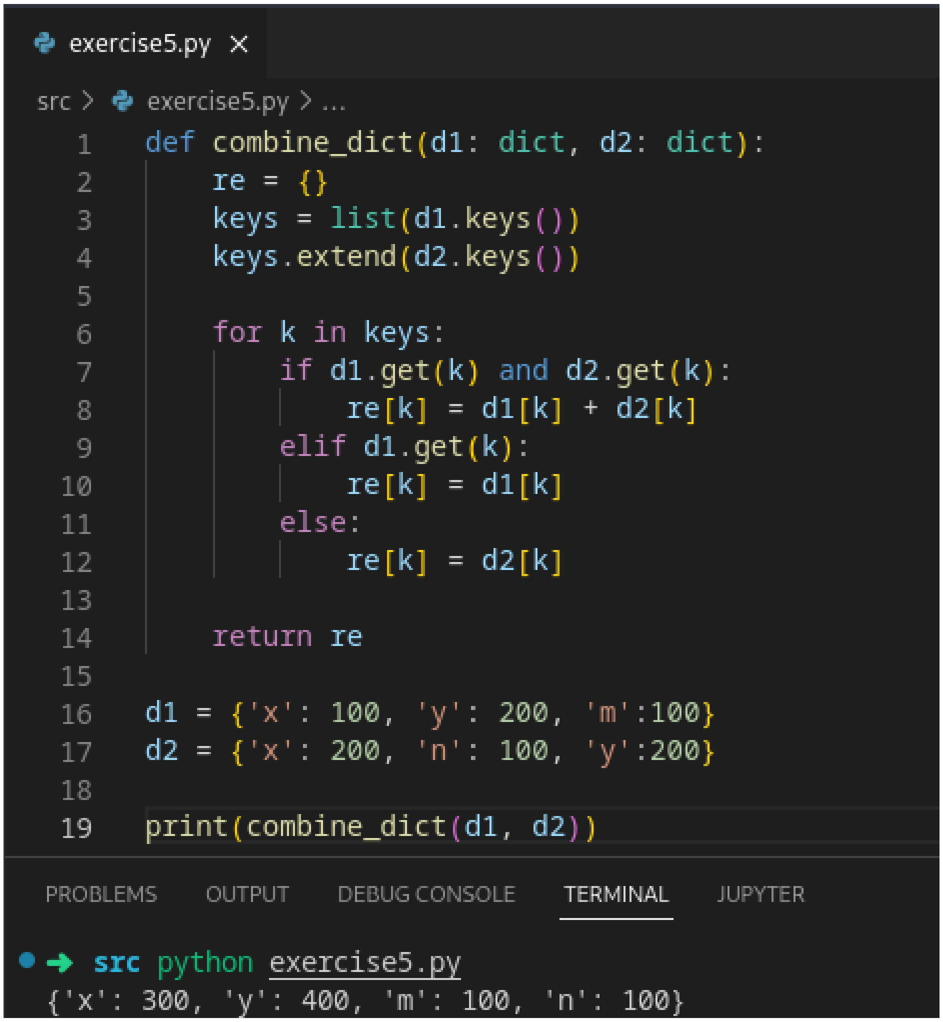
```python
def word_frequencies(string):
    punctuation = [',', '.', ':', ';', '\'', '"']
    s = "".join(i for i in string if i not in punctuation)
    re = {}
    for i in s.split(" "):
        if re.get(i):
            re[i] += 1
        else:
            re[i] = 1
    return re

s = "Fred fed Ted bread, and Ted fed Fred bread"

print(word_frequencies(s))
```

```
➜ src python exercise4.py
{'Fred': 2, 'fed': 2, 'Ted': 2, 'bread': 2, 'and': 1}
```

Exercise-5:

Write a Python program to combine two dictionaries, adding values for common keys.

```
d1 = {'x': 100, 'y': 200, 'm':100}
d2 = {'x': 200, 'n': 100, 'y':200}
```



```python
def combine_dict(d1: dict, d2: dict):
    re = {}
    keys = list(d1.keys())
    keys.extend(d2.keys())

    for k in keys:
        if d1.get(k) and d2.get(k):
            re[k] = d1[k] + d2[k]
        elif d1.get(k):
            re[k] = d1[k]
        else:
            re[k] = d2[k]

    return re

d1 = {'x': 100, 'y': 200, 'm':100}
d2 = {'x': 200, 'n': 100, 'y':200}

print(combine_dict(d1, d2))
```

```
● ➜ src python exercise5.py
{'x': 300, 'y': 400, 'm': 100, 'n': 100}
```