Solution Review: Keys Matching in Multiple Dictionaries

This lesson will explain how to find keys matching a given value with in as a parameter to the function within multiple dictionaries.

```
WE'LL COVER THE FOLLOWING
Solution 1: Use a Nested for Loop
Solution 2: Use a for Loop
```

Solution 1: Use a Nested for Loop

- The outer for loop iterates over the outer dictionary using students.items()
- The inner for loop iterates over the subdictionary values using subdict.values()
- If the subitem matches with the address in the parameter, the key is saved in a list. Return the sorted key using sorted(list).

The following python code demonstrates the concept:

```
def find_students(address, students):
    names = []
    for key, subdict in students.items():
        for sublist in subdict.values():
            if (sublist == address):
                names.append(key)
    return sorted(names)

students = {
        "Peter": {"age": 10, "address": "Lisbon"},
        "Isabel": {"age": 11, "address": "Sesimbra"},
        "Anna": {"age": 9, "address": "Lisbon"},
}
print(find_students("Lisbon", students))
```







Solution 2: Use a for Loop

- A for loop iterators over the dictionary using key and value to keep track of items of the dictionary.
 - If the value of the address in the dictionary is equal to the given address the corresponding keys get appended to names.
 - The sorted names are returned

```
def find_students(address, students):
    names = []
    for key, value in students.items():
        if value["address"] == address:
            names.append(key)
        return sorted(names)

students = {
        "Peter": {"age": 10, "address": "Lisbon"},
        "Isabel": {"age": 11, "address": "Sesimbra"},
        "Anna": {"age": 9, "address": "Lisbon"},
    }
    print(find_students("Lisbon", students))
```

Now that you have an insight on dictionaries, let's move on to the quiz before moving on to the next chapter.