```python
# https://codeshare.io/ez86YK
# Dictionary Exercises # SaiPrabath Chowdary S

'''
Exercise 1: Create a Dictionary
1. Create a dictionary called `person` with the following key-value pairs:
   - Name: "Alice"
   - Age: 25
   - City: "New York"
2. Print the dictionary.
'''

person = {"Name": "Alice", "Age": 25, "City": "New York"}
print(person)

'''
Exercise 2: Access Dictionary Elements
1. Access the value of the `"Name"` key in the `person` dictionary and print it.
2. Access the value of the `"City"` key and print it.
'''
print(person["Name"])
print(person["City"])

'''
Exercise 3: Add and Modify Elements
1. Add a new key-value pair to the `person` dictionary: `"email": "alice@example.com"`.
2. Change the value of the `"Age"` key to 26.
3. Print the modified dictionary.
'''
person["email"] = "alice@example.com"
person["Age"] = 26
print(person)

'''
Exercise 4: Remove Elements
1. Remove the `"City"` key from the `person` dictionary.
2. Print the dictionary after removing the key.
'''
del person["City"]
print(person)
```

```python
'''
Exercise 5: Check if a Key Exists
1. Check if the key `"email"` exists in the `person` dictionary. Print a message
based on the result.
2. Check if the key `"phone"` exists in the dictionary. Print a message based on
the result.
'''
if "email" in person:
    print("email key exists")
else:
    print("email key does not exist")

if "phone" in person:
    print("phone key exists")
else:
    print("phone key does not exist")

'''
Exercise 6: Loop Through a Dictionary
1. Iterate over the `person` dictionary and print each key-value pair.
2. Iterate over the keys of the dictionary and print each key.
3. Iterate over the values of the dictionary and print each value.
'''
for key, value in person.items():
    print(key, value)

for key in person.keys():
    print(key)

for value in person.values():
    print(value)

'''
Exercise 7: Nested Dictionary
1. Create a dictionary called `employees` where the keys are employee IDs (`101`,
`102`, `103`) and the values are dictionaries containing employee details (like
name and job title). Example structure:
   employees = {
       101: {"name": "Bob", "job": "Engineer"},
       102: {"name": "Sue", "job": "Designer"},
       103: {"name": "Tom", "job": "Manager"}}
2. Print the details of employee with ID `102`.
```

```python
3. Add a new employee with ID `104`, name `"Linda"`, and job `"HR"`.
4. Print the updated dictionary.
'''
employees = {
    101: {"name": "Bob", "job": "Engineer"},
    102: {"name": "Sue", "job": "Designer"},
    103: {"name": "Tom", "job": "Manager"}
}
print(employees[102])
employees[104] = {"name": "Linda", "job": "HR"}
print(employees)


'''
Exercise 8: Dictionary Comprehension
1. Create a dictionary comprehension that generates a dictionary where the keys
are numbers from 1 to 5 and the values are the squares of the keys.
2. Print the generated dictionary.
'''
squares = {x: x**2 for x in range(1, 6)}
print(squares)

'''
Exercise 9: Merge Two Dictionaries
1. Create two dictionaries:
   dict1 = {"a": 1, "b": 2}
   dict2 = {"c": 3, "d": 4}
2. Merge `dict2` into `dict1` and print the result.
'''
dict1 = {"a": 1, "b": 2}
dict2 = {"c": 3, "d": 4}
dict1.update(dict2)
print(dict1)




'''
Exercise 10: Default Dictionary Values
1. Create a dictionary that maps letters to numbers: `{"a": 1, "b": 2, "c": 3}`.
2. Use the `get()` method to retrieve the value of key `"b"`.
3. Use the `get()` method to try to retrieve the value of a non-existing key
`"d"`, but provide a default value of `0` if the key is not found.
'''
```

```python
letter_map = {"a": 1, "b": 2, "c": 3}
print(letter_map.get("b"))
print(letter_map.get("d", 0))



'''
Exercise 11: Dictionary from Two Lists
1. Given two lists:
   keys = ["name", "age", "city"]
   values = ["Eve", 29, "San Francisco"]
2. Create a dictionary by pairing corresponding elements from the `keys` and
`values` lists.
3. Print the resulting dictionary.
'''
keys = ["name", "age", "city"]
values = ["Eve", 29, "San Francisco"]
person_info = dict(zip(keys, values))
print(person_info)

'''
Exercise 12: Count Occurrences of Words
1. Write a Python program that takes a sentence as input and returns a dictionary
that counts the occurrences of each word in the sentence.
   sentence = "the quick brown fox jumps over the lazy dog the fox"
2. Print the dictionary showing word counts.
'''
sentence = "the quick brown fox jumps over the lazy dog the fox"
words = sentence.split()
word_count = {word: words.count(word) for word in set(words)}
print(word_count)
```