**1.** The email\_list function receives a dictionary, which contains domain names as keys, and a list of users as values. Fill in the blanks to generate a list that contains complete email addresses (e.g. diana.prince@gmail.com).

def email\_list(domains):

    emails = []

    for domain, users in domains.items():

      for user in users:

        emails.append(user + "@" + domain)

    return(emails)

print(email\_list({"gmail.com": ["clark.kent", "diana.prince", "peter.parker"], "yahoo.com": ["barbara.gordon", "jean.grey"], "hotmail.com": ["bruce.wayne"]}))

**2.**The groups\_per\_user function receives a dictionary, which contains group names with the list of users. Users can belong to multiple groups. Fill in the blanks to return a dictionary with the users as keys and a list of their groups as values.

def groups\_per\_user(group\_dictionary):

    user\_groups = {}

    # Go through group\_dictionary

    for group, users in group\_dictionary.items():

        # Now go through the users in the group

        for user in users:

            # Now add the group to the the list of

    return(user\_groups)

print(groups\_per\_user({"local": ["admin", "userA"],

            if user in user\_groups:

                user\_groups[user].append(group)

            else:

                user\_groups[user] = [group]

            # groups for this user, creating the entry

            # in the dictionary if necessary

        "public": ["admin", "userB"],

        "administrator": ["admin"]}))

3.The add\_prices function returns the total price of all of the groceries in the dictionary. Fill in the blanks to complete this function.

def add\_prices(basket):

    # Initialize the variable that will be used for the calculation

    total = 0

    # Iterate through the dictionary items

    for item, price in basket.items():

        # Add each price to the total calculation

        # Hint: how do you access the values of

        # dictionary items?

        total += basket[item]

    # Limit the return value to 2 decimal places

    return round(total, 2)

groceries = {"bananas": 1.56, "apples": 2.50, "oranges": 0.99, "bread": 4.59,

    "coffee": 6.99, "milk": 3.39, "eggs": 2.98, "cheese": 5.44}

print(add\_prices(groceries)) # Should print 28.44