A logo with blue and red text

Description automatically generated A logo with text on it

Description automatically generated

**Introduction to Programming**

**Lab Worksheet**

**Week 7**

**Python**

Name: Shivnath Mahato

Level 4 Section: A

British Id: 10012

LBU Id: c7576124

Level 4 BSc. Hons Computing

Subject: Fundamental of Computer Programming

(FOCP)

The British College (TBC)

**Task:**

**#Week 7 Practical**

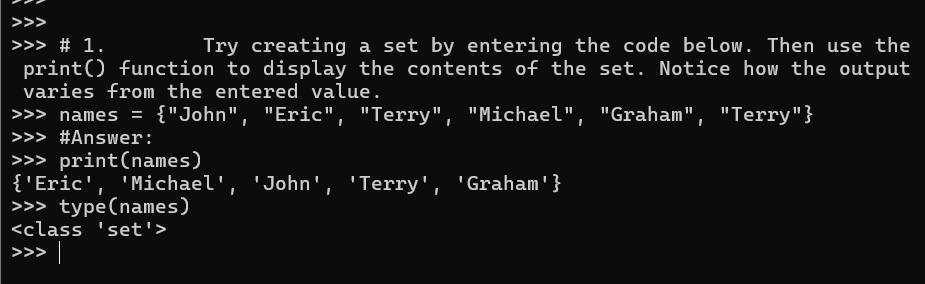
1. Try creating a set by entering the code below. Then use the print() function to display the contents of the set. Notice how the output varies from the entered value.

names = {"John", "Eric", "Terry", "Michael", "Graham", "Terry"}

**Answer:**

print(names)

**Output of Question No. 1:**

****

1. Enter the code below, then make a call to the print() function to display the contents of the set.

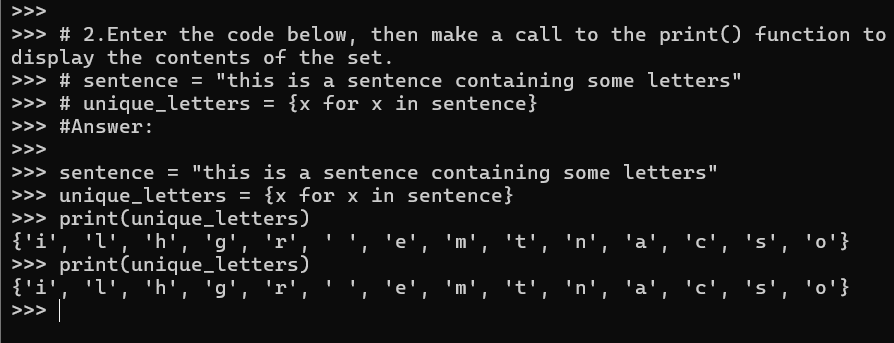
sentence = "this is a sentence containing some letters"

unique\_letters = {x for x in sentence}

**Answer:**

print(unique\_letters)

**Output of Question No. 2:**



1. Rewrite the previous code so that it checks that the input name is NOT within the set of known names. Hint: use the not in operator.

names = {"John", "Eric", "Terry", "Michael", "Graham", "Terry"}

**Answer:**

input\_name = input("Enter a name: ")

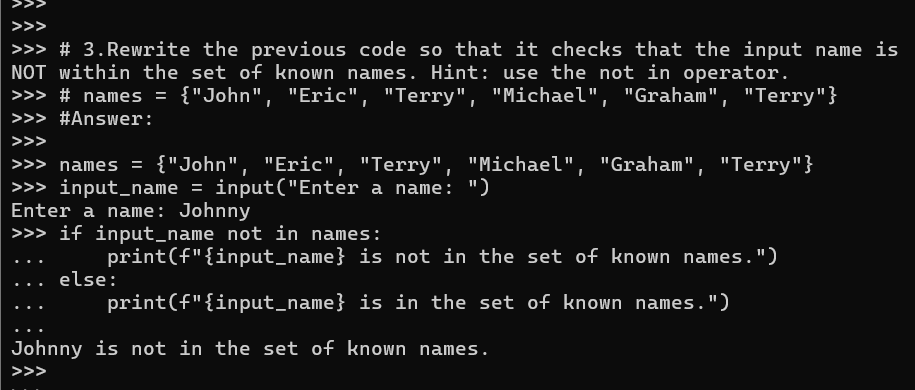
if input\_name not in names:

    print(f"{input\_name} is not in the set of known names.")

else:

    print(f"{input\_name} is in the set of known names.")

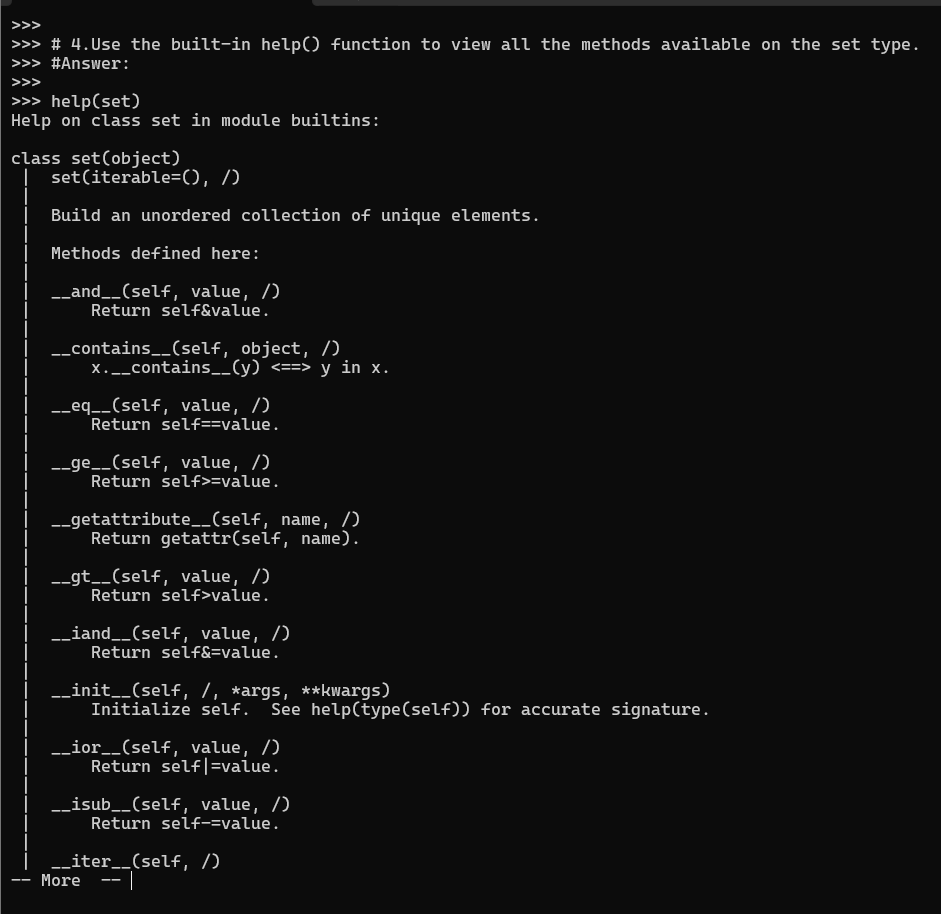
**Output of Question No. 3:**



1. Use the built-in help() function to view all the methods available on the set type. **Answer:**

help(set)

**Output of Question No. 4:**



1. Create the two initial sets, staff and directors as shown in the first example above. Perform the four mathematical set operations shown, but use the equivalent *method* calls to achieve the same results. For example:

staff = staff | {"Mark", "Ringo"}

# becomes …

staff = staff.union({"Mark", "Ringo"})

**Answer:**

# Union

union\_set = staff.union(directors)

print("Union:", union\_set)

# Intersection

intersection\_set = staff.intersection(directors)

print("Intersection:", intersection\_set)

# Difference

difference\_set = staff.difference(directors)

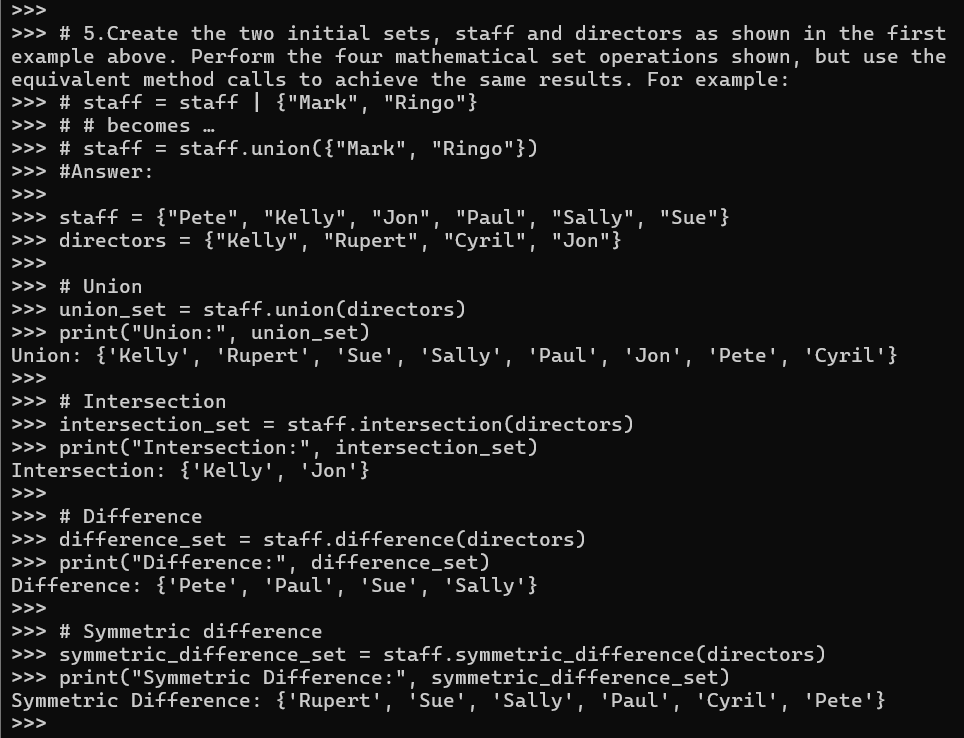
print("Difference:", difference\_set)

# Symmetric difference

symmetric\_difference\_set = staff.symmetric\_difference(directors)

print("Symmetric Difference:", symmetric\_difference\_set)

**Output of Question No. 5:**



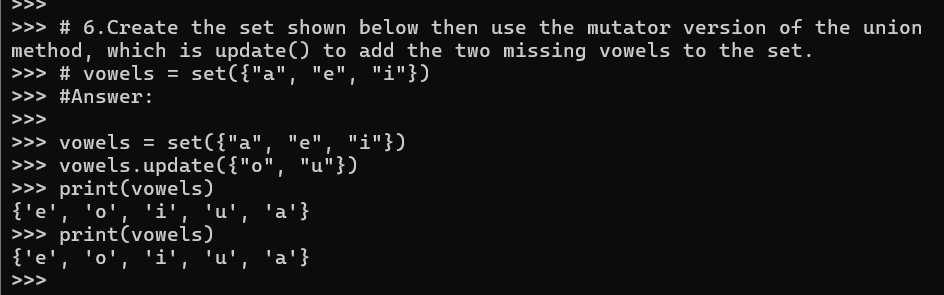
1. Create the set shown below then use the *mutator* version of the union method, which is update() to add the two missing vowels to the set.

vowels = set({"a", "e", "i"})

**Answer:**

vowels.update({"o", "u"})

print(vowels)

**Output of Question No. 6:**

1. Write code based on the previous two examples, but use the equivalent *method* calls to achieve the same results.

staff = {"Pete", "Kelly", "Jon", "Paul", "Sally", "Sue"}

directors = {"Kelly", "Rupert", "Cyril", "Jon"}

**Answer:**

#Union

union\_set = staff | directors

print("Union: ",union\_set)

# Intersection

intersection\_set = staff & directors

print("Intersection:", intersection\_set)

# Difference

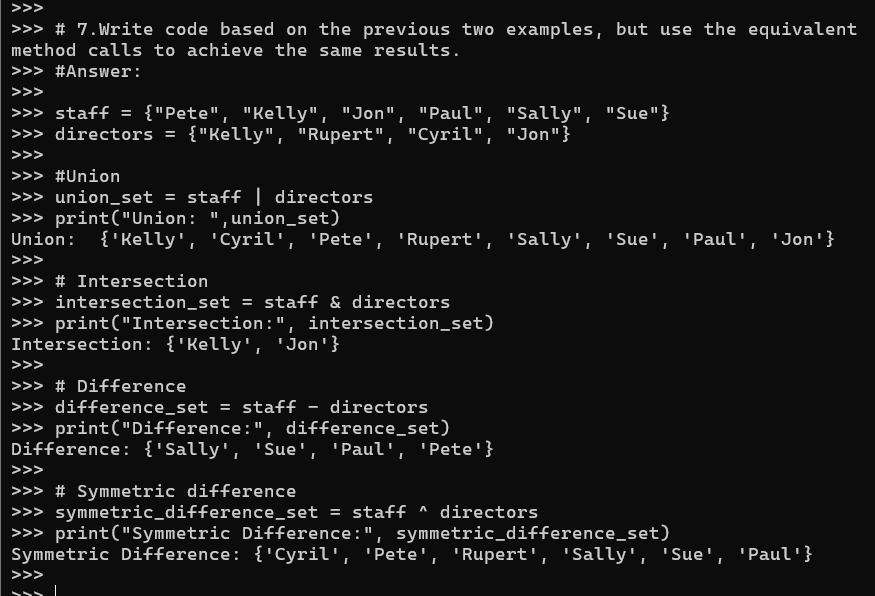
difference\_set = staff - directors

print("Difference:", difference\_set)

# Symmetric difference

symmetric\_difference\_set = staff ^ directors

print("Symmetric Difference:", symmetric\_difference\_set)

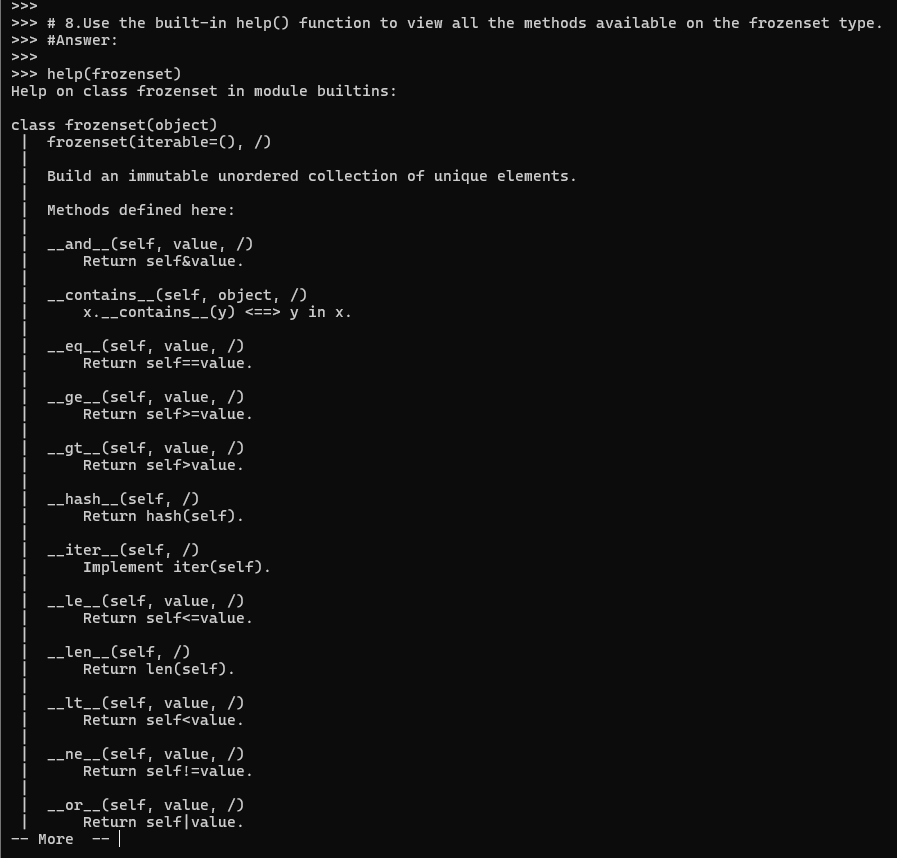
**Output of Question No. 7:**

1. Use the built-in help() function to view all the methods available on the frozenset type.

**Answer:**

help(frozenset)

**Output of Question No. 8:**



1. Enter the code below, then make a call to the print() function to display the contents of the dictionary.

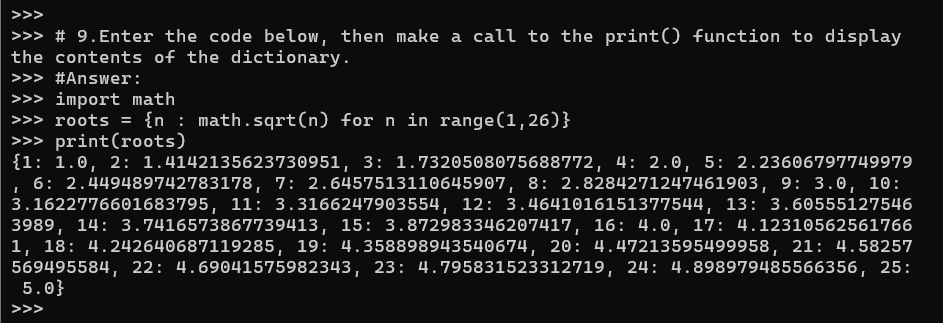
import math

roots = {n : math.sqrt(n) for n in range(1,26)}

**Answer:**

print(roots)

**Output of Question No. 9:**



1. Write some code which adds a new fruit called "kiwi" to the stock dictionary, with an initial stock level of 10.

stock = {"apple":10, "banana":15, "orange":11}

stock["kiwi"] = 10 # add new key:value pair

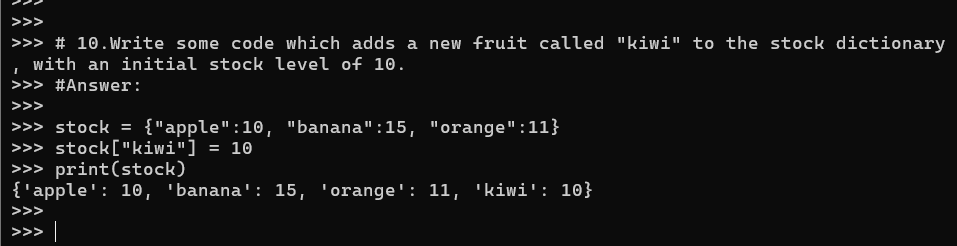
**Answer:**

stock = {"apple":10, "banana":15, "orange":11}

stock["kiwi"] = 10

print(stock)

**Output of Question No. 10:**

****

1. Use the built-in help() function to view all the methods available on the dict type. Then write some code that uses the popitem() method to remove some *key:value* pairs from the stock dictionary.

stock = {"apple":10, "banana":15, "orange":11}

**Answer:**

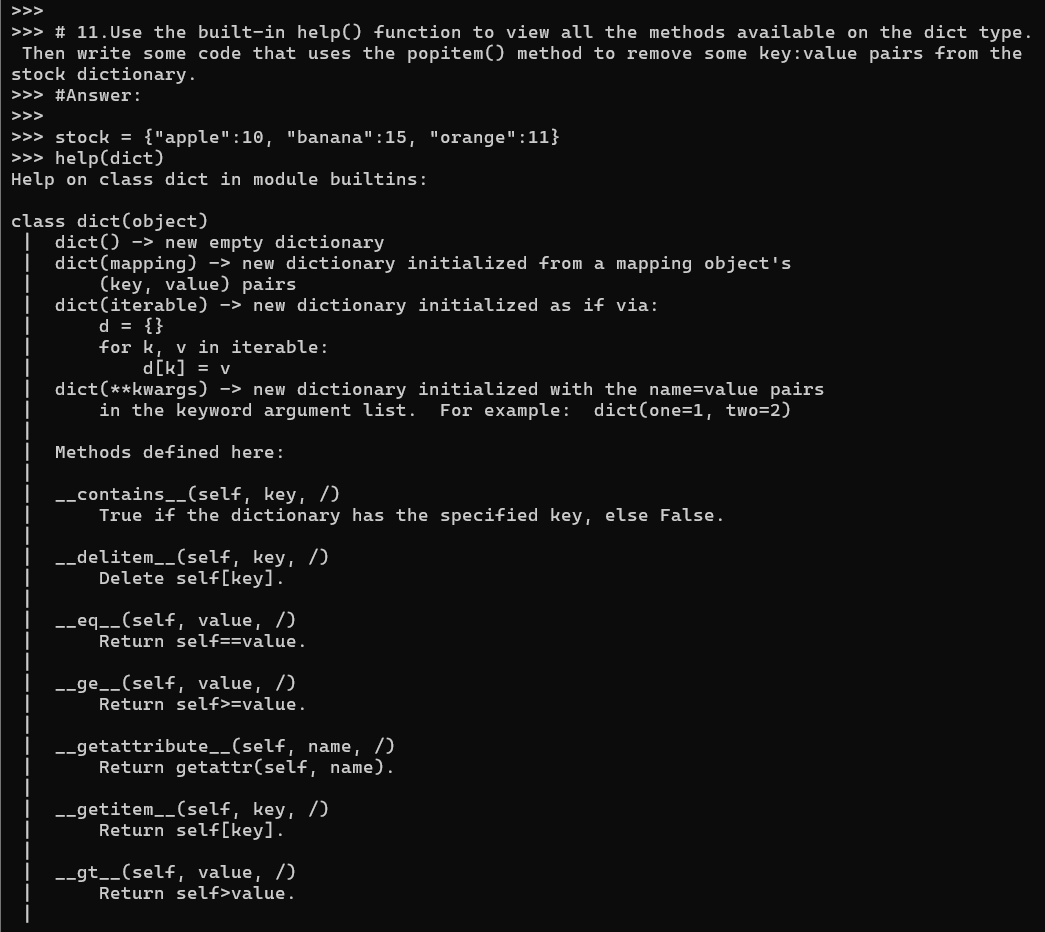
help(dict)

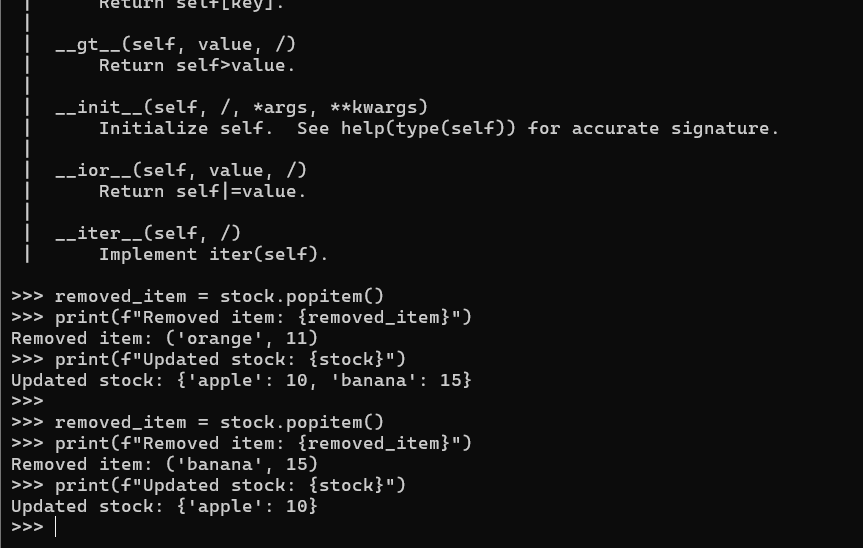
removed\_item = stock.popitem()

print(f"Removed item: {removed\_item}")

print(f"Updated stock: {stock}")

**Output of Question No. 11:**

****

****

1. Write some code that iterates over the contents of the roots dictionary created within an earlier task. For each entry, print the message -

“The square root of <num> is <sqrt>”

**Answer:**

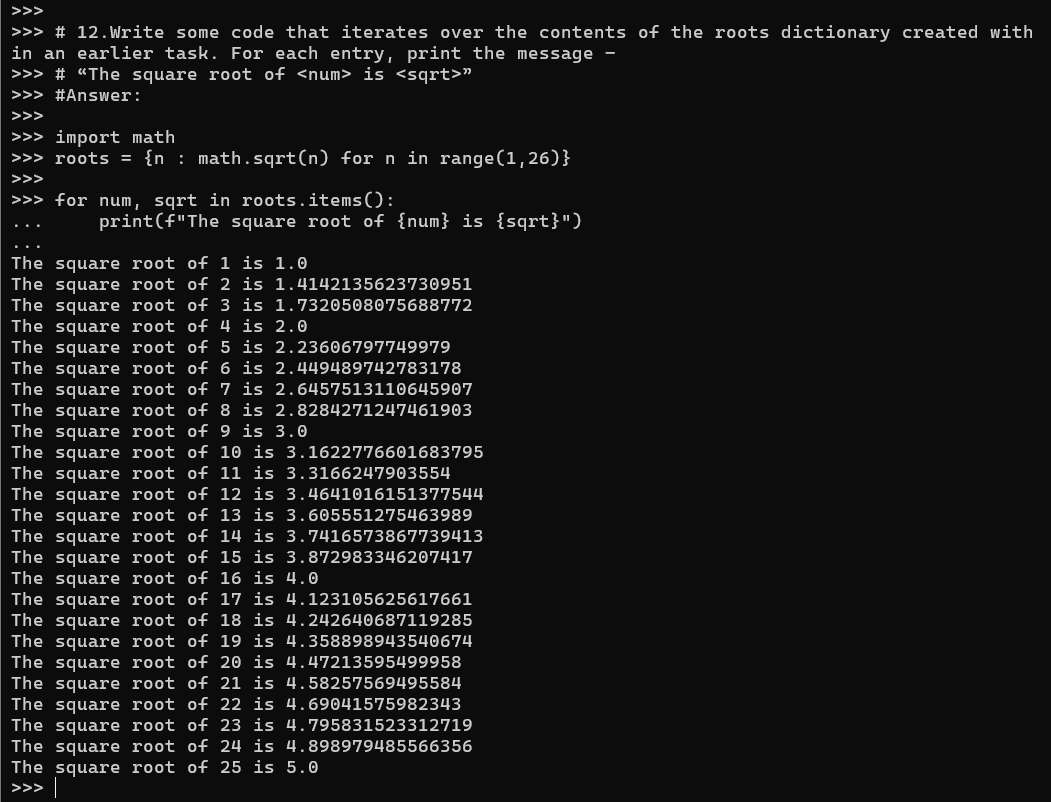
import math

roots = {n : math.sqrt(n) for n in range(1,26)}

for num, sqrt in roots.items():

    print(f"The square root of {num} is {sqrt}")

**Output of Question No. 12:**



1. Look at each of the phrases below and ensure you understand what each of these means. For any that you do not understand, do a little research to find a definition of each term. This research may involve looking back over these notes, or the associated lecture notes. It may also involve searching for these terms on the Internet.

* Set
* Set operations
* Set comprehension
* Dictionary
* Key:value pair

**Answer:**

* **Set:** A set is an unordered collection with no duplicate elements.
* **Set operations:** Set operations include union, intersection, difference, and symmetric difference.
* **Set comprehension:** Set comprehension is a method for creating sets in python using the elements from other iterables like lists, sets, or tuples.
* **Dictionary:** Dictionary is a data structure which is ordered, mutable and stores elements as pairs often called key:value pair
* **Key:value pair:** 
  + **Key**: A unique identifier that is used to reference the corresponding value.
  + **Value:** The data associated with the key.

Key can not be duplicate while value can.