QUEENSBOROUGH COMMUNITY COLLEGE
The City University of New York
Department of Engineering Technology

Programming Exercises - Dictionaries

1. Determine the output displayed by the lines of code. Save your code as *PE8_1.py*.
   NY gives the populations of the five boroughs in millions.

```
NY = {"BX":1.42, "MN":1.63, "QS":2.25, "BN":2.56, "SI": 0.47}
```

A	`print((NY['QS']))`
	`print(NY.get("QS"))`
		B	`print(NY.get("LI", "Not in"))`
			`print(NY.get('SI', 'absent'))`
			`print(NY.setdefault('SI', 0.48))`

Output		Output	

C	`print("LI" in NY)`
	`print('MN' not in NY)`
		D	`print(len(NY), min(NY), max(NY))`
			`print(len(NY.items()),`
			`max(NY.keys()), min(NY.values()))`

Output		Output	

E	`print(round(NY['QS']))`
	`NY['QS'] += .3`
	`print(round(NY['QS'], 1))`
		F	`print(NY.keys())`
			`print(list(NY.values()))`
			`print(tuple(NY.items()))`

Output		Output	

G	`total = 0`
	`for x in NY.values():`
	`  total += x`
	`print(f'{total:.1f}')`
		H	`total = 0`
			`for x in NY:`
			`  total += NY[x]`
			`print(f'{total:.1f}')`

Output		Output	
I	`for x in sorted(NY) : print(x, end = '|')`		
Output			
J	Use a for loop to print all key names in the reversed alphabetical order (see output below).		
Output	SI|QS|MN|BX|BN|		
K	Use a for loop to print all values from max to min order (see output below).		
Output	2.56, 2.25, 1.63, 1.42, 0.47,		
L	`if "QS" in NY: print("Queens is the most diverse county in NY.")`		
Output			
M	`for x, y in NY.items():` `  if y > 2.5: print(f"{x} is the Kings county!")`		
Output			
N	`NY["SK"] = 1.49` `print(NY)`	O	`NY.update({"NU":1.34})` `print(NY)`
Output		Output	

P	`NY.pop("QS")`
	`NY.popitem()`
	`print(NY)`
		Q	`newYork = NY`
			`del newYork['BN']`
			`print(NY)`
			`print(newYork)`

Output		Output	

R	`newYork = dict(NY)`
	`del newYork["BN"]`
	`print(len(NY))`
	`print(len(newYork))`
		S	`NewYork = NY.copy()`
			`NY.clear()`
			`print(NY)`
			`print(NewYork)`
			`del NY`
			`print(set(NewYork))`

Output		Output	

2. Convert the following two lists into one dictionary:

```
keys = ['Ten', 'Twenty', 'Thirty']
values = [10, 20, 30]

Example Output:
{'Ten': 10, 'Twenty': 20, 'Thirty': 30}
```

3. Merge the following two dictionaries into one dictionary:

```
dict1 = {'Ten': 10, 'Twenty': 20, 'Thirty': 30}
dict2 = {'Thirty': 30, 'Forty': 40, 'Fifty': 50}

Example Output:
{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Forty': 40, 'Fifty': 50}
```

4. Create a dictionary and use loops to print keys and values:
a) Create a dictionary, *stuInfo* with the keys *name*, *gpa*, and *age*. Give appropriate values for each key.
b) Use a loop and the *items()* method to print all keys and values.
c) Use the *update()* method to change the gpa to 4.0.
d) Use a loop and the *keys()* method to print all keys and values.
e) Add a key *major* with the value to the dictionary.
f) Use a loop and the *values()* method to print all values.
g) Use two different ways to delete *gpa* and *age* in the dictionary.
h) Print the updated dictionary.

Example Output:

```
NAME    John Smith
GPA     3.456
AGE     20

NAME    John Smith
GPA     4.0
AGE     20

John Smith|4.0|20|CSIS|

{'name': 'John Smith', 'major': 'CSIS'}
```

5. Displays a rank in the defined dictionary.
a) Create a dictionary, rank = {1:"Freshman", 2:"Sophmore", 3:"Junior", 4:"Senior"}
b) Request a user input for a number of years.
c) Print the value of the matching key in the dictionary.
d) Print the error message if input is invalid.

Example Output 1

```
Enter the # of years in the school (1-4): 1
Year 1 = Freshman
```

Example Output 2

```
Enter the # of years in the school (1-4): 11
Invalid years.
```

6. Nest dictionaries within a list, stuClass.
a) Create three *stuInfo* dictionaries with the keys: *name* and value: *gpa*. Add appropriate values for each key.
b) Create a list *stuClass*, add all dictionaries to this list, and print the list.
c) Use a loop to print all *students* from the list of *stuClass.*
d) Use a loop to print all the *gpa.*
e) Change the last student's *gpa* to 4.0.
f) Add a new student info to the list.
g) Use a loop to print all the names and *gpa* with proper format as the output below.

Example Output

```
All students in the list:
[{'name': 'tom cat', 'gpa': 3.456}, {'name': 'jerry mouse', 'gpa': 4.0}, {'name': 'sponge bob', 'gpa': 3.99}]

All students information:
student 1 {'name': 'tom cat', 'gpa': 3.456}
student 2 {'name': 'jerry mouse', 'gpa': 4.0}
student 3 {'name': 'sponge bob', 'gpa': 3.99}

All gpa information:
3.456|4.0|3.99|

All the updated information:
Tom Cat             3.46
Jerry Mouse         4.00
Sponge Bob          4.00
John Smith          3.99
```

7a. Implement the following:
a) Use a for loop to create a list with 26 letters (a-z).
b) Use a for loop to create a list with 26 numbers from 1 to 26 inclusive.
c) Create a dictionary, *charNum* by zipping above two lists.
d) Use a for loop to print the keys and values in the dictionary as the output below.

Example Output:

```
a 1|b 2|c 3|d 4|e 5|f 6|g 7|h 8|i 9|j 10|k 11|l 12|m 13|n 14|o 15|p 16|q 17|r 18|s 19|t 20|u 21|v 22|w 23|x 24|y 25|z 26|
```

7b. Implement the following:
a) Use a for loop to create a list with 26 letters (A-Z).
b) Use a for loop to create a list with 26 numbers from 100 to 2600 (with step value of 100) inclusive.
c) Create a dictionary, *numChar* by zipping above two lists.
d) Use a for loop to print the keys and values in the dictionary as the output below.
e) Merge the dictionary, *charNum* created in 7a with the dictionary, *numChar* into one dictionary, *all*.
f) Print out the dictionary, *all* as the output below.

Example Output:

```
100 A|200 B|300 C|400 D|500 E|600 F|700 G|
```
…
```
|2100 U|2200 V|2300 W|2400 X|2500 Y|2600 Z|
```

```
{'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5, 'f': 6, 'g': 7, 'h': 8, 'i': 9, 'j': 10, 'k': 11,
```
…

```
1800: 'R', 1900: 'S', 2000: 'T', 2100: 'U', 2200: 'V', 2300: 'W', 2400: 'X', 2500: 'Y', 2600: 'Z'}
```