Activity 3
Collections and Iteration: Looping Through Data
Applied Python Programming with AI and Raspberry Pi Interfaces

Instructor: Dr. Vikas Thammanna Gowda   Semester: ABCD 20YX

Assign: *TBD*

Name: _____

1. True or False

Statement	T/F
The `break` statement can be used to exit a loop prematurely.	
It is possible to use the `break` statement to exit multiple nested loops at once.	
The `continue` statement is used to skip the remaining code within the current iteration of a loop and move to the next iteration.	
The sequence generated from `range(n)`, where `n` is an integer, starts at 1 and goes up to (but does not include) `n`.	
Nested loops require using both `for` and `while` loops.	
Infinite loops occur when the condition controlling the loop never becomes false or when there is no exit statement within the loop body.	
The body of a `while` loop in Python is always guaranteed to execute at least once.	
A `while True` loop will run indefinitely until a `break` statement is encountered or an error occurs within the body.	
The loop condition in a `while` loop is evaluated before each iteration.	
The loop control variable in loops like `for x in range(2)` can be accessed outside the loop.	
Sets do not allow duplicates	
Dictionaries allow duplicate keys	
Dictionaries allow duplicate values	
Tuples are mutable	
Lists are ordered	

2. What is the output of the following code snippets?

a.

```python
num = 4
result = 1
i = 1
while (i <= num):
    result *= i
    i += 1
print("The result is:", result)
```

Output:

b.

```python
start_num = 6
while (start_num > 0):
    print(start_num)
    start_num -= 1
```

Output:

c.

```python
num = 10
while (num > 10):
    if ((num % 2) == 0):
        print("Hello from if!")
    else:
        print("Hello from else!")
```

Output:

d.

```
result = 0
for i in range(start, limit + 1, 2):
    result += i
print("The result is:", result)
```

Output:

Case 1: `start = 1, limit = 10`

Case 1: `start = 2, limit = 11`

e.

```
count = 1
while (count <= 10):
    if (count % 3 == 0):
        count += 2
    print(count, end=" ")
    count += 1
```

Output:

f.

```
value_x = 0
while value_x < 5:
    value_x += 1
    if value_x == 3:
        continue
    print(value_x)
```

Output:

g.

```python
passwords= ["1234", "admin", "guest", "letmein"]
for pwd in passwords: # pwd is a variable
    print("Checking:", pwd)
    if pwd == "guest":
        print("Password found!")
        break
```

Output:

h.

```python
scores = [89, -1, 76, 92, -5, 83]
for s in scores:
    if s < 0:
        continue
    print("Valid score:", s)
```

Output: