Problem 1.

process_age:

a) $\forall$ age, age $< 0 \Rightarrow$ "Invalid"

$\forall$ age, $0 \leq$ age $< 18 \Rightarrow$ "Minor"

$\forall$ age, age $> 65 \Rightarrow$ "Senior"

$\forall$ age, $18 \leq$ age $\leq 65 \Rightarrow$ "Adult"

b)

```
assert process_age(-5) == "Invalid"
assert process_age(10) == "Minor"
assert process_age(70) == "Senior"
assert process_age(32) == "Adult"
```

c) 4 Test cases needed for 100% branch coverage.

Sum_until_negative

a)

path 1: $len(numbers) = 0 \Rightarrow total = 0$

path 2: $len(numbers) > 0 \wedge \forall i \in [0, n-1], numbers[i] \geq 0$

$$\Rightarrow total = \sum_{i=0}^{n-1} numbers[i]$$

path 3: $\exists j \in [0, n-1], numbers[j] < 0 \wedge \forall i \in [0, j-1], numbers[i] \geq 0$

$$\Rightarrow total = \sum_{i=0}^{j-1} numbers[i]$$

path 4: $number[0] < 0 \Rightarrow total = 0$.

b)

```
assert sum_until_negative([]) == 0
assert sum_until_negative([1, 2, 3]) == 6
assert sum_until_negative([1, -3, 4]) == 1
assert sum_until_negative([-1]) == 0
```

c) 4 test cases needed for 100% branch coverage.

classify_sequence:

(a) Assuming an input array of size 3

Path 1: all three numbers are positive ⇒ "AllPositive"

Path 2: all three numbers are non-positive ⇒ "AllNonpositive"

path 3: A mix of positive and non-positive numbers ⇒ "Mixed"

(b)

len(numbers) = 0 ⇒ "Empty"

len(numbers) > 0 ∧ for ∀ i, numbers[i] > 0

⇒ "Allpositive"

len(numbers) > 0 ∧ numbers[j] < 0 for ∃ j ∈ [0, n-1] ∧ numbers[i] for ∀ i ≠ j

⇒ "Mixed"

len(numbers) > 0 ∧ numbers[i] < 0 for ∀ i

⇒ "AllNonpositive.

(c) python code attached. www.github.com/Jamiezoomies/ucla-cs130

(d) assert classify_sequence([])

assert classify_sequence([1,2,3])

assert classify_sequence([-1,-5,-4])

assert classify_sequence([1,2,-1])

(e) Empty array and large array ($n > 5$) input cases.

problem 2.

www.github.com/Jamiezoomies/ucla-cs130

q2_cloud_alert.py