

The mini-test is to be completed in the **Python workspace available to you in Codio (Seminar Mini-Test)** where you will need to provide the coding scripts that fulfil the required outcomes. A sample code to help guide you is available in the Python workspace.

1. Code a function called 'first_div_16' ACCEPT two positive integers, n1 and n2, as inputs RETURN the first number in range(n1,n2) that is divisible by 16. HOWEVER, if no number in the range is divisible by 16 RETURN 0.

```
#Question1
def first_div_16(n1, n2):
    for i in range(n1, n2):
        if i % 16 == 0:
            return i
    return 0
```

2. Code a function called 'halve_to_2' ACCEPT one numeric input. If the number <= 0, RETURN -1. If the number > 0, divide that integer over-and-over by 2 until it becomes smaller than 2. RETURN that smaller-than-2 number, e.g. input of 4 Will yield 1 (4->2->1), 5 yields 1.25 (5->2.5->1.25) etc.

```
#Question2
def halve_to_2(num):
    if num <= 0:
        return -1
    while num >= 2:
        num /= 2
    return num
```

3. Code a function called 'string_expansion'. ACCEPT a non-empty string as input RETURN a string that contains every other character, 2n+2 times, where n is the original index of the letter. e.g. Input of "Hello" should result in "HHllllllloooooooooo". Input of "ROBErt" should result in "RRBBBBBBrrrrrrrrrr".

```
#Question3
def string_expansion(s):
    result = ""
    for i in range(len(s)):
        result += s[i] * (2 * i + 2)
    return result
```

4. Code a function called 'item_count_from_index'. ACCEPT two inputs, a list and an integer-index RETURN a count (number) of how many times the item at that index appears in the list. HOWEVER, if the integer-index is out of bounds for the list RETURN the empty string ("") (e.g. list of 3 items, index of 5 is out of bounds).

#Question4

```
def item_count_from_index(lst, index):
    if index < 0 or index >= len(lst):
        return ""
    count = 0
    for item in lst:
        if item == lst[index]:
            count += 1
    return count
```

5. Code a function called 'length_times_largest'. ACCEPT a list as input RETURN the length of the list times the largest integer (not float) in the list. HOWEVER, if the list does not contain an integer, RETURN the empty string ("").

#Question5

```
def length_times_largest(lst):
    integers = [i for i in lst if isinstance(i, int)]
    if len(integers) == 0:
        return ""
    else:
        return len(lst) * max(integers)
```