Problem Solving using Python

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.

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

# Example usage of first_div_16 function
result = first_div_16(10, 50)
print(result)

result = first_div_16(50, 100)
print(result)

result = first_div_16(100, 120)
print(result)

result = first_div_16(200, 210)
print(result)
```

```
16
64
112
208
```

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.

```
def halve_to_2(number):
    if number <= 0:
        return -1

    while number >= 2:
        number /= 2

    return number

# Example usage of halve_to_2 function
result = halve_to_2(4)
print(result)

result = halve_to_2(5)
print(result)

result = halve_to_2(10)
print(result)

result = halve_to_2(0)
print(result)

result = halve_to_2(-5)
print(result)
```

```
1.0
1.25
1.25
-1
-1
```

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 "HHlllllloooooooooo". Input of "ROBErt" should result in "RRBBBBBBrrrrrrrrr".

```python
def string_expansion(input_string):
    result = ''

    for index, char in enumerate(input_string):
        if index % 2 == 0:
            result += char * (2 * index + 2)

    return result

# Example usage of string expansion function
result = string_expansion("Hello")
print(result)

result = string_expansion("ROBErt")
print(result)
```

```
HHlllllloooooooooo
RRBBBBBBrrrrrrrrr
```

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).

```
def item_count_from_index(input_list, index):
    if index < 0 or index >= len(input_list):
        return ""

    item = input_list[index]

    return input_list.count(item)

# Example usage of item_count_from_index function
result = item_count_from_index([1, 2, 3, 1, 2, 1], 2)
print(result)

result = item_count_from_index([1, 2, 3, 1, 2, 1], 4)
print(result)

result = item_count_from_index([1, 2, 3, 1, 2, 1], 5)
print(result)

result = item_count_from_index([1, 2, 3, 1, 2, 1], 6)
print(result)
```

1
2
3
Not Found “”

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 ("").

```
def length_times_largest(input_list):
    integers = [item for item in input_list if isinstance(item, int)]

    if not integers:
        return ""

    largest_integer = max(integers)

    return len(input_list) * largest_integer

# Example usage of length_times_largest function
result = length_times_largest([1, 2, 3, 4, 5])
print(result)

result = length_times_largest([2, 3, 'a', 'b'])
print(result)

result = length_times_largest([1.5, 'hello', 3.0])
print(result)

result = length_times_largest([7, 1, 3, 2])
print(result)
```

25
12
Not Found “”
28