Problem 3 - Python Stack Traces

Challenge: Python Stack Trace Interpretation
See the "Python Stack Traces" attachment which lists several python stack traces. Your task is to examine the stack traces and provide a brief response for each one that summarizes what the problem or likely problem is, and the first line of code you would jump to in your code editor given the trace.

```
Traceback Problem 1
==================
Traceback (most recent call last):
  File "stack_traces.py", line 36, in run_trace
    f()
  File "stack_traces.py", line 45, in <lambda>
    run_trace(1, lambda: perform_calculation(add, '1', 3))
  File "stack_traces.py", line 8, in perform_calculation
    calc(x, y)
  File "stack_traces.py", line 12, in add
    return x + y
TypeError: can only concatenate str (not "int") to str
```

- A way to tackle this problem would be to either change the definition of the add function in line 12 to sanitize inputs or to change the parameters passed in line 45 to have two values that can be “added”

```
Traceback Problem 2
==================
Traceback (most recent call last):
  File "stack_traces.py", line 36, in run_trace
    f()
  File "stack_traces.py", line 46, in <lambda>
    run_trace(2, lambda: perform_calculation(add, 7, '3'))
  File "stack_traces.py", line 8, in perform_calculation
    calc(x, y)
  File "stack_traces.py", line 12, in add
    return x + y
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

- This is the same issue as above, except this time on line 46, the two values can’t be added together, 7 and ‘3’ could be changed to ‘7’ and ‘3’ if the intention is to concatenate or 7 and 3 if the intention is to add (likely the case from func name).

Traceback Problem 3
==========
```
Traceback (most recent call last):
  File "stack_traces.py", line 36, in run_trace
    f()
  File "stack_traces.py", line 47, in <lambda>
    run_trace(3, lambda: perform_calculation(mult, '3', '3'))
  File "stack_traces.py", line 8, in perform_calculation
    calc(x, y)
  File "stack_traces.py", line 15, in mult
    return x * y
TypeError: can't multiply sequence by non-int of type 'str'
```

- The issue here is with the mult function (or the values passed to it), right now in line 47 two strings are being passed which can't be multiplied, the solution could be to change them to numbers 3, 3 instead of '3','3'

Traceback Problem 4
==========
```
Traceback (most recent call last):
  File "stack_traces.py", line 36, in run_trace
    f()
  File "stack_traces.py", line 48, in <lambda>
    run_trace(4, lambda: perform_calculation(mult, [4], [3]))
  File "stack_traces.py", line 8, in perform_calculation
    calc(x, y)
  File "stack_traces.py", line 15, in mult
    return x * y
TypeError: can't multiply sequence by non-int of type 'list'
```

- The issue here is the same as in the trace above, the mult function can't take two operands of type list so they could be changed to 4,3

Traceback Problem 5
==========
```
Traceback (most recent call last):
  File "stack_traces.py", line 36, in run_trace
    f()
  File "stack_traces.py", line 49, in <lambda>
    run_trace(5, lambda: perform_calculation(innoc, '1', 3))
  File "stack_traces.py", line 8, in perform_calculation
    calc(x, y)
  File "stack_traces.py", line 22, in innoc
    spelunk()
```

```
  File "stack_traces.py", line 21, in spelunk
    raise ValueError('Invalid')
ValueError: Invalid
```

- The issue here is with the innoc function, in line 22, it calls a second function called spelunk which just raises a ValueError of ‘Invalid’

Traceback Problem 6

```
==================
Traceback (most recent call last):
  File "stack_traces.py", line 36, in run_trace
    f()
  File "stack_traces.py", line 50, in <lambda>
    run_trace(6, lambda: comp_calc([1, 2, 3], 1, add))
  File "stack_traces.py", line 30, in comp_calc
    return [perform_calculation(calc, x_i, y_i) for x_i, y_i in zip(x, y)]
TypeError: zip argument #2 must support iteration
```

- The issue here is with the values passed to comp_calc in line 30 (definition) and 50 (where the values are passed). The zip function requires 2 values which support iteration the list [1,2,3] does but element 1 doesn’t

Traceback Problem 7

```
==================
Traceback (most recent call last):
  File "stack_traces.py", line 36, in run_trace
    f()
  File "stack_traces.py", line 51, in <lambda>
    run_trace(7, lambda: comp_calc([1, 2, [3]], [4, 5, 6], add))
  File "stack_traces.py", line 30, in comp_calc
    return [perform_calculation(calc, x_i, y_i) for x_i, y_i in zip(x, y)]
  File "stack_traces.py", line 30, in <listcomp>
    return [perform_calculation(calc, x_i, y_i) for x_i, y_i in zip(x, y)]
  File "stack_traces.py", line 8, in perform_calculation
    calc(x, y)
  File "stack_traces.py", line 12, in add
    return x + y
TypeError: can only concatenate list (not "int") to list
```

- The issue here is a bit of a compound of the issues before, the line to check is line 51, where the inputs should be changed to [1,2,3] and [4,5,6]

Traceback Problem 8

```
==================
```

```
Traceback (most recent call last):
  File "stack_traces.py", line 36, in run_trace
    f()
  File "stack_traces.py", line 52, in <lambda>
    run_trace(8, lambda: calc_dict({'one': 1, 'two': '2'}, 'one', 'two', add))
  File "stack_traces.py", line 26, in calc_dict
    return perform_calculation(calc, d[k1], d[k2])
  File "stack_traces.py", line 8, in perform_calculation
    calc(x, y)
  File "stack_traces.py", line 12, in add
    return x + y
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

- The issue is in line 52 with the dictionary passed to calc_dict, the second element is ‘two’:’2’ the function will try to add the values for keys ‘one and ‘two’ which are 1 ,’2’ bringing us back to the same issue as trace 1 above

```
Traceback Problem 9
==================
Traceback (most recent call last):
  File "stack_traces.py", line 36, in run_trace
    f()
  File "stack_traces.py", line 53, in <lambda>
    run_trace(9, lambda: calc_dict({}, 'one', 'two', add))
  File "stack_traces.py", line 26, in calc_dict
    return perform_calculation(calc, d[k1], d[k2])
KeyError: 'one'
```

- The issue is with line 53, the calc_dict function will try to access the first element passed with keys ‘one’ and ‘two’, the issue is that the dictionary is empty so accessing key ‘one’ is not possible