Task 1:

Write a function called `squareInteger()` that uses input function to prompt user for a positive integer and prints its square. Use error checking to handle cases were input is not positive integer.

For example:

Please enter a positive integer: two

Your input is not understandable. Try again.

Please enter a positive integer: 2.5

Your input is not understandable. Try again.

Please enter a positive integer: -2

Your input is not a positive integer. Try again.

Please enter a positive integer: 2

Your number squared is 4

The signature of this function is `(None) ---> None`.

Task 2:

For this task, write a function, `combine_lists(list1, list2)` that combines two lists into a dictionary. The elements of the first one serve as keys and the elements of the second one serve as values. We will assume that each element in the first list is unique, and that they are all valid dictionary keys.

Example:

Input:
```
['a', 'b', 'c', 'd', 'e']
[1, 2, 3, 4, 5]
```

Output:
```
{'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}
```

The signature of this function is `(list, list) ---> dict`.

In the example above, the arguments to the function are both of the same length. However, users often do nonsensical things, and we might get two lists of different lengths. Use a conditional statement ("if block") to make sure that both lists are of the same length, and raise an exception (`ValueError` would be reasonable) if they're not. In addition, it's possible that the first list contains duplicates, which means we don't have enough keys to match to our values. Raise a `KeyError` if your code detects this situation in the provided arguments.

Then, *outside* of the function, where you're calling it, use a `try`/`except` block to try to call the function, and then catch the exceptions that (might) result, and printout an informative error message for each one.