1. What is the relationship between def statements and lambda expressions?

- They both are of same type 'Function'.
- Lambda definition does not include a "return" statement, it always contains an expression which is returned. We can also put a lambda definition anywhere a function is expected, and we don't have to assign it to a variable at all.
- While using def, we needed to define a function with a name and need to pass a value to it. After execution, we also needed to return the result from where the function was called using the *return* keyword.

2. What is the benefit of lambda?

- The lambda keyword provides a shortcut for declaring small anonymous functions. Lambda functions behave just like regular functions declared with the def keyword. They can be used whenever function objects are required.
- We can also put a lambda definition anywhere a function is expected, and we don't have to assign it to a variable at all

3. Compare and contrast map, filter, and reduce.

- Map creates a new array by transforming every element in an array, individually.
- Filter creates a new array by removing elements that do not belong.
- Reduce, takes all the elements in an array and reduces them into a single value.

  Based on the usage we can conclude that

- If you want to apply function to each element of iterator then use MAP
- If you want to filter elements of iterator by some condition then use FILTER
- If you do some computation like addition, product on iterator's elements and want single value in output then use REDUCE.

4. What are function annotations, and how are they used?

- Function annotations are a Python 3 feature that lets you add arbitrary metadata to function arguments and return value.
- Function annotations are completely optional both for parameters and return value.
- Function annotations provide a way of associating various parts of a function with arbitrary python expressions at compile time.

5. What are recursive functions, and how are they used?

- A recursive function is a function which will continue to call itself and repeat its behaviour until some condition is met to return a result.
- Recursive functions in code often rely on loop setups, where the initial variable is called on multiple times while being altered by the loop.

6. What are some general design guidelines for coding functions?

- Use the function naming rules lowercase with words separated by underscores as necessary to improve readability.
- Use one leading underscore only for non-public methods and instance variables.
- To avoid name clashes with subclasses, use two leading underscores to invoke Python's name mangling rules.

7. Name three or more ways that functions can communicate results to a caller.

- When the task is carried out, the function can or cannot return one or more values. There are three types of functions in Python: Built-in functions, such as help () to ask for help, min() to get the minimum value, print() to print an object to the terminal,… You can find an overview with more of these functions here.