Consider the following fragment of Python code:

```
import pandas as pd

column_names = ["col1", "col2", "col3"]
values = [("a", 0, 1),
  ("a", 1, 1),
  ("b", 0, 0)]

d = pd.DataFrame(values, columns=column_names)

def mysterious_function(val):
 if val == 0: return val * 7 + 1
 return val * 2

d["col4"] = d["col3"].apply(mysterious_function)

d.groupby("col1").col4.sum()
```

1. Qualitatively explain what the program does and describe the expected output.

The code creates a pandas DataFrame using the three tuples in `values` *as rows.*

A fourth column is added applying `mysterious_function` *to the values of column* `col3`*. The function returns 1 if value is 0, else it returns the double of value.*

Finally, the records are grouped by `col1` *values and the values of* `col4` *in each group are summed, leading to:*

```
a 4
b 1
```

2. Implement an alternative mysterious_function that returns a list of all the integers from 0 to val included.

```
def mysterious_function(val):
 return list(range(0, val + 1))
```

3. Choose a pure Python data structure to store the output of the code execution, and motivate your choice

After grouping, `col1` *values are unique, each one corresponding to the sum of* `col4` *calculated per group. A Python dictionary would be a natural choice for key / value pairs.*