## Part A: Dictionary Utilities

The exercises in parts B and C make use of dictionaries whose keys are strings and whose values are numbers. In this section you are asked to develop some functions that will be useful in processing such dictionaries:

### Exercise 1 (1 mark):

Write a function getTotal whose parameter is a dictionary whose values are numeric, and which returns the sum of those values.

Example:

>>> total = getTotal({'A':3,'B':5,'C':2})

>>> total

10

### Exercise 2 (1 mark):

Write a function normalise whose parameter is a dictionary whose values are numeric, and which returns a dictionary that has the same keys, but where the values have been divided by their sum (as returned by the function described above) so that they add up to 1.

Example

>>> norm = normalise({'A':3,'B':5,'C':2})

>>> norm

{'A': 0.3, 'B': 0.5, 'C': 0.2}

### Exercise 3 (1 mark):

Write a function printNonZero whose parameter is a dictionary with numeric values and which prints out those key-value pairs where the value is non-zero, one pair per line.

Example:

>>> mydict = {'A':3,'B':5,'C':0,'D':2}

>>> printNonZero(mydict)

A : 3

B : 5

D : 2

### Exercise 4 (2 marks):

Write a function analyse that takes two parameters: The first is a dictionary with keys representing parties, and values representing the votes that party got in an election; the second is a dictionary with the same keys but where the values are the numbers seats that each party was allocated. The function should print out a comparison of the percentage of votes that each party got, versus the percentages of seats. The function should only output details for parties getting at least one seat. The percentages should be rounded to the nearest integer. **Hint:** You may find that the normalise function is useful here.

Example:

>>> votes = {'A':112,'B':52,'C':2,'D':36}

>>> seats = {'A':3,'B':1,'C':0,'D':1}

>>> analyse(votes,seats)

A : 55 % of votes vs 60 % of seats

B : 25 % of votes vs 20 % of seats

D : 17 % of votes vs 20 % of seats

### Exercise 5 (2 marks):

Write a function addTo which takes as parameters two dictionaries (we will refer to them as d1 and d2) both of which have numeric values. After calling the function d2 is unchanged, but d1 is modified as follows:

* If a key is present in d2 and not in d1, then it is added to d1, and associated with the value it has in d2.
* If a key is present in both dictionaries, then the value of that key in d1 is set to be the sum of the values that it had in the two dictionaries.

Example

>>> myd1 = {'A':3,'B':1,'C':6,'D':1}

>>> myd2 = {'X':5,'A':1,'Y':3,'D':7}

>>> addTo(myd1,myd2)

>>> myd1

{'A': 4, 'B': 1, 'C': 6, 'D': 8, 'X': 5, 'Y': 3}