SOFTWARE 1 PRACTICAL

FUNCTIONS

Week 4 – Practical 3

Exercise 1:

Write a **function** `sum_digits(number)` to calculate and return the sum of the digits of a given whole number (int) given as parameter. For example,

```
>>> print(sum_digits(1234))
10
```

Exercise 2:

Write a **function** `pairwise_digits(number_a, number_b)` that takes two integers as parameters and returns a binary string where a character 1 is used if the digits at the same index are the same, a 0 otherwise. Examples are given in the table below.

Input A	Input B	Output
1213	2113	‘0011’
1213	10435	‘10010’
1213	121	‘1110’

Exercise 3:

Write a **function** `to_base10(binary)` that take a binary number (base 2), convert it into a decimal number (base 10) and return the base 10 value. To compute such a value, we need to understand what a binary number is.

Index	7	6	5	4	3	2	1	0
Binary	1	0	0	0	1	0	1	1
Decimal 139	1×2^7	0×2^6	0×2^5	0×2^4	1×2^3	0×2^2	1×2^1	1×2^0
	128	0	0	0	8	0	2	1

The binary number 10001011 represents the number 139, whereas the number 11111111 represents 255.

Exercise 4:

Write a python script to print a right angle triangle composed of alternating 0s and 1s.. For example:

```
>>> Input number of rows: 5
1
01
101
0101
10101
```

Exercise 5: *Where's that bug!*

You have just started your placement, and you are given a piece of code to correct. The aim of the script is to take a 2D list (that is a list of lists) and print a list containing the sum of each list. For example, given the list in `data`, the `output` should be `[6, 2, 10]`.

Modify the code below such that it gives the right result. In addition, you have been asked to refactor the script into a **function** `sum_lists(list_2D)` that returns the list containing the sums of each sub-list.

```
data = [[1,2,3], [2], [1, 2, 3, 4]]
output =[]
total = 0

for row in data:
    for val in row:
        total += val
        output.append(total)

print(output)
```