SOFTWARE 1 PRACTICAL

ITERATION

Week 3 – additional exercises

Exercise 1:

1. Write a script asking the user to enter a series of positive integers separated by a white space, and then prints the number of even numbers that was entered. For example:

```
>>> enter a series of numbers: 1 2 4 3 6 8 5 2 11 100 101
There are 6 even numbers
```

   You will need to use the built-in function `input()` and the string method `split()`. You should also remember that we can convert a string representing an integer into an int like so:

```
>>> twenty = int('20')
>>> type(twenty)
<class 'int'>
```

2. Same question except we would like to have the list of even number as well

```
>>> enter a series of numbers: 1 2 4 3 6 8 5 2 11 100 101
There are 6 even numbers: 2 4 6 8 2 100
```

3. Same question except we would like to have the list of even number without duplicate, that is remove the second 2 in the example below.

```
>>> enter a series of numbers: 1 2 4 3 6 8 5 2 11 100 101
There are 5 distinct even numbers: 2 4 6 8 100
```

Exercise 2:

The aim of this exercise is to draw a sudoku board (not to solve it). To start with we will be using a 4 × 4 board (as opposed to the classic 9 × 9).

1. Write a script that asks a user to enter 4 digits (comprised between 0 and 4) separated by a white space, the store each digit in a list and print the list.

```
>>> enter 4 digits (0..4) separated by a space: 0 2 1 4
[0, 2, 1, 4]
```

2. Write a script that ask a user to enter 4 times a sequence of 4 digits, and store it in a 2D list, that is a list containing 4 lists of 4 digits each. The script should print the 2D list. The output should look like:

```
[[0,2,1,4],[3,4,2,1],[1,2,3,4],[0,0,2,3]]
```

3. Modify your script so the output looks likes:

```
+-+-+-+-+
|0|2|1|4|
+-+-+-+-+
|3|4|2|1|
+-+-+-+-+
|1|2|3|4|
+-+-+-+-+
|0|0|2|3|
+-+-+-+-+
```

4. Modify your script so the 0 are replaced by a blank space.

```
+-+-+-+-+
| |2|1|4|
+-+-+-+-+
|3|4|2|1|
+-+-+-+-+
|1|2|3|4|
+-+-+-+-+
| | |2|3|
+-+-+-+-+
```

Exercise 3: *from "Python Programming Fundamental"- Kent L, 2nd Ed. (2014)*

There is an elegant algorithm for converting a decimal number to a binary number. You need to carry out long division by 2 to use this algorithm. If we want to convert 83_{10} to binary then we can repeatedly perform long division by 2 on the quotient of each result until the quotient is zero. Then, the string of the remainders that were accumulated while dividing make up the binary number. For example,

```
83/2 = 41 remainder 1
41/2 = 20 remainder 1
20/2 = 10 remainder 0
10/2 = 5 remainder 0
5/2 = 2 remainder 1
2/2 = 1 remainder 0
1/2 = 0 remainder 1
```

The remainders from last to first are 1010011_2 which is 83_{10}. This set of steps is called an algorithm. An algorithm is like a recipe for doing a computation. We can use this algorithm any time we want to convert a number from decimal to binary.

Implement a script that prompts the user to enter a positive in and print a binary representation of that number.