Problem Solving Methodology in IT (COMP1001)

Assignment Six
(Due at noon on 1 Nov 2018)

(Instructions: see the instructions in Blackboard)

Rocky K. C. Chang

24 October 2018

1. [Weight = 4] (Formatting the printout) Write a Python program that will print a range of numbers in rows and columns. The range is specified by user. Moreover, user is asked to input the number of numbers printed on each row and the number of spaces for printing each number. Each row is filled up from left to right, and a new row will be used only after a row is filled up. Each number is printed on right-justified format.

   A sample output:

```
Please enter a starting number and an ending number: 1, 31
Please enter the number of numbers printed on a row
and the number of spaces per item: 10, 4
--------------------------------------

Your print-out of numbers 1-31 using 10 columns
and 4 spaces between numbers:

   1    2    3    4    5    6    7    8    9   10
  11   12   13   14   15   16   17   18   19   20
  21   22   23   24   25   26   27   28   29   30
  31
>>> main()
Please enter a starting number and an ending number: 32, 70
Please enter the number of numbers printed on a row
and the number of spaces per item: 6, 8
--------------------------------------

Your print-out of numbers 32-70 using 6 columns
and 8 spaces between numbers:

      32      33      34      35      36      37
      38      39      40      41      42      43
      44      45      46      47      48      49
      50      51      52      53      54      55
      56      57      58      59      60      61
      62      63      64      65      66      67
      68      69      70
```

   Hint: There are two challenges to this question. The first is to determine when to print a new row. The key is after printing **every** *n* (which is provided by user) numbers, a new line begins. The second is to provide *m* spaces for printing each number. Please reference to the example just above section 7.1.4 in https://docs.python.org/3.1/library/string.html.

2. [Weight = 4] (Binary number addition) In this question we continue Q2 of A4. In that question, we have

```
tableAdd = ("00", "01", "10")

def PROC0(digit1, digit2):
    """
    Input: digit1, digit2, each of which is either 0 or 1.
    Output: a string of two digits which is the sum of the two digits.
    Therefore, the result could be "00", "01" or "10".
    """
    return tableAdd[digit1+digit2]

def PROC1(digit1, digit2, digit3):
    """
    Input: digit1, digit2, digit3, each of which is either 0 or 1.
    Output: a string of two digits which is the sum of the three digits.
    Therefore, the result could be "00", "01", "10", or "11".
    """
    firstPROC0Result = PROC0(digit1, digit2)
    secondPROC0Result = PROC0(int(firstPROC0Result[1]),digit3)
    if firstPROC0Result[0] == secondPROC0Result[0] == "0":
        result = "0"
    else:
        result = "1"

    return result+secondPROC0Result[1]
```

In this question you will implement *PROC2()* with the function signature below. Notice that this function signature is not the same as what we have covered in the class. You must use *PROC1()* in your implementation. Recall that we have developed the pseudocode in the slides on Computation. You may use *int()* to remove the leading 0 in a string of a number.

Function PROC2(num1, num2)
   Input: *num1 and num2 are two multidigit binary numbers.* Their lengths do not have to be the same.
   Output: *The result of the binary addition of num1 and num2.*

Include the statements below in your .py file.

```
print(PROC2(0,1))
print(PROC2(1,1))
print(PROC2(0,0))
print(PROC2(101,1))
print(PROC2(111,111))
print(PROC2(10101010,1111100))
```