THE UNIVERSITY OF THE WEST INDIES
Department of Computing
COMP1126–Introduction to Computing I

Lab 4

Exercise

Type these expressions in the python shell.

```
>>>[] + [1,2]
>>>[1,2]+[3]
>>>5/2.0
>>>5//2.0
>>>len(['apples','oranges',True,4])
>>>[x for x in range (0,10) if x % 3 == 0 ]
>>> [x for x in range (0,10) if x % 2 == 0 or x % 5 == 0]
>>>len([x for x in range (0,10) if x % 2 == 0])
>>>[(x,y,2010) for x in range (1,8) for y in range(1,13)]
```

Include these lists in your code.

```
month_days= [('January',[31]),('February',[28,29]),('March',[31]),
('April',[30]),('May',[31]),('June',[30]),('July',[31]),('August',[31]),
('September',[30]),('October',[31]),('November',[30]),('December',[31]) ]

day_names =
['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']
```

Problem 1

Write a function `days_in_month` which takes a month as an argument and finds the corresponding month in the `month_days` list and returns the number of days associated with that month.

```
>>> days_in_month('December')
[31]
>>> days_in_month('February')
[28, 29]
```

Problem 2

Zeller's Congruence is an algorithm for finding the day of the week for any date. Zeller's formula is as follows:

day = (((13*m+3) // 5 + d + y + (y // 4) - (y // 100) + (y // 400)) %7)

where
d = day, y = year and m = month

Note: If the month is January or February then you add 12 to the month and subtract 1 from the year before calculating the day.

The result is a day number in the range 0..6 where the corresponding day can be extracted from the `day_names` list by using an appropriate index.

e.g. `day_names[0] = 'Monday' and day_names[6] = 'Sunday'.`

Define a python function `day_of_week`, which displays the day name for a given date supplied in the form (day,month,year).
e.g.

```
>>> day_of_week(18,10,2016)
'Tuesday'
>>> day_of_week(21,10,2016)
'Friday'
>>> day_of_week(20,10,2016)
'Thursday'
```

Problem 3

Using list comprehension, define a python function `unlucky`, which returns all the days in a given year which have the date Friday 13th e.g.

```
>>> unlucky(2010)
[(13, 8, 2010)]
>>> unlucky(2015)
[(13, 2, 2015), (13, 8, 2015), (13, 11, 2015)]
```

[Hint: you need two ranges one for day starting from 1 and going to 31 and another one for month starting from 1 going to 12. Using these and the year which comes as an argument and use the function `day_of_week` in the if part of list comprehension to check if a given date is 'Friday' and also check if the day is equal to 13.]

Problem 4

Write a python function `mostUnlucky`, which lists all the years between 0 and 2016 which have 3 unlucky days. Use function `unlucky` to get a list of unlucky dates for a particular year and find the length of this list. If the length is greater than 2 then the year is added to another list which is returned as output. Write this function twice (give different names), once using list comprehension and another iteratively (i.e. using for loops).