15110 PRINCIPLES OF COMPUTING – LABEXAM 2- Summer 2015

Name: ____________________Section: _______Andrew Id: _______________Machine:_____

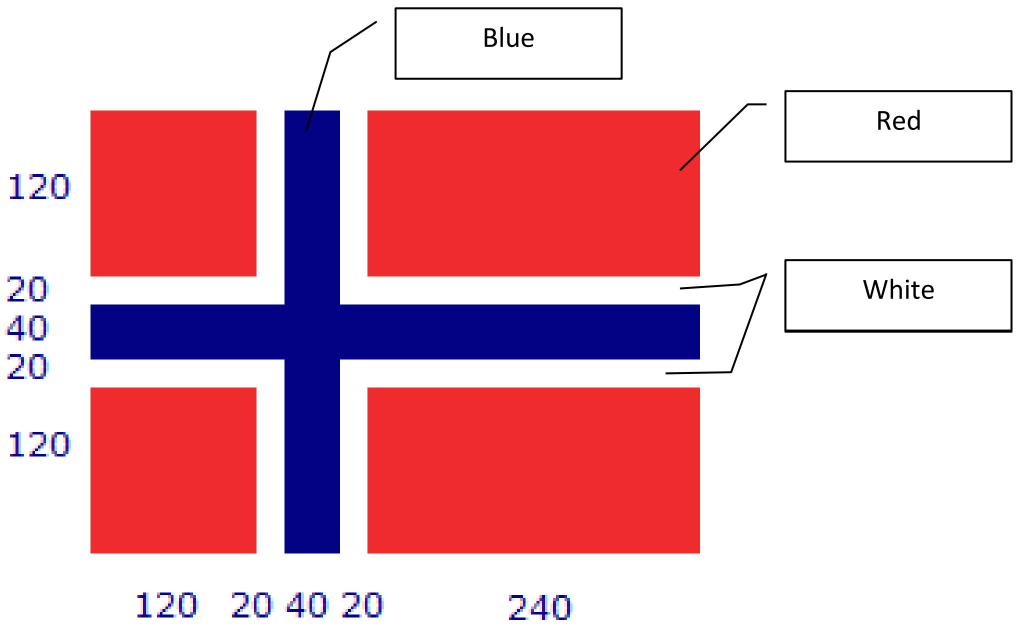
Directions:

1. In your home directory, **create a folder** named **labexam2**
2. Write a function in Python for each of the following problems and store these functions in the labexam2 folder. **Test your functions** by running in IDLE or calling them with python3 –i. Although we give you example/test runs, your function should work on all legal inputs based on the specifications given, and your output should match the examples as closely as possible for full credit. Remember that we will run your code on additional test cases that are not shown on the exam.
3. Once you are finished, **compress the labexam2 folder into a zip file and submit** it to **AutoLab** (http://autolab.cs.cmu.edu) by the end of lab. **Do not delete the labexam2 folder** from your home directory.

Question 1 (save the file as q1.py in your labexam2 folder) [25 points]

Write a Python function NorwayFlag() that displays the flag of Norway in a window of size 440 by 320 pixels. The flag consists of a red background with two blue and white stripes placed on each other. The lengths are given in the image below. Your result should look like the flag on the right.

Hint: You may think it in different ways such as "Drawing red background first" or "Drawing a blue background first". Or you may consider drawing rectangles and lines next to each other (this option might be longer and takes more time)



Question 2 (save the file as q2.py in your labexam2 folder) [25 points]

Write a function peakCount(data) that takes a list of integers as input, and returns the number of peak values. A peak value is a number which is bigger than both of its predecessor and successor. For example in the list [2, 3, 3, 2, **4**, 1, 8, **9**, 1, 5] the peak values are 4 and 9. You can assume that the first and last numbers in the list cannot be the peak values.

Use the following algorithm:

1. Create a variable *count* and initialize it.
2. For each number in the list which has a predecessor and a successor.
    a. Check if the number is a peak number or not and if it is a peak number increase the count by one
3. return *count* of peaks.

Sample usage:	

>>> a = [1,2,3,2,4,3,5,6,6]
>>> peakCount(a)
2
	>>> b = [23,20, 20, 25, 25, 19,20,10, 13, 5, 6, 2]
	>>> peakCount(b)
	3

>>> peakCount([1,2,3]) 0	>>> peakCount([ ]) 0

Question 3 (save the file as q3.py in your labexam2 folder) [25 points]

Write a function rowSums(matrix) that takes a matrix (an array of arrays) of integers and prints the sums of each row in a different line as shown in sample runs below

Hint: Trace the matrix for each row and columns to get the values but calculate the sum for each row.

Sample usage:	

>>> data = [[3,5,2],[2,7,4],[2,5,9]]
>>> rowSums(data)
Sum of row 1 : 10
Sum of row 2 : 13
Sum of row 3 : 16
	>>> data2 = [[10, 20, 30, 40], [11, 22, 33, 44]]
	>>> rowSums(data2)
	Sum of row 1 : 100
	Sum of row 2 : 110

Question 4 (save the file as q4.py in your labexam2 folder) [25 points]

Write a **recursive** function digitSum(number) that takes a positive integer as an input and returns the sum of its digits. **Do not use a for/while loop**; you must use recursion.

Sample usage:	
>>> digitSum(1234) 10	>>> digitSum(135789) 33
>>> digitSum(12) 3	>>> digitSum(532255341) 30