CSC241 – Lab 5

Instructions: Submit a single file lab5.py containing solutions to the following problems. Make sure you run the doctest to check your solutions before you submit. Copy and paste the results of the doctest in a multiline comment at the top of your submission.

1. Write a function `doubles` that takes as a list of integers as a parameter and prints the values in the list, one per line, that are exactly twice the previous value in the list. If the list is empty or there are no values that are double the value of their predecessor, then the function does not print anything. The list provided as a parameter should not be changed. Note that the first item in the list will never be printed since it does not have a predecessor in the list. Sample usage:

```
>>> doubles( [4,8,-12,-24,48,3,6,12,24,2])
8 -24 6 12 24
>>> doubles([1,2,3,4,5])
2
>>> doubles([])
>>>
```

2. Write a function `oddSpots()` that takes as a list of integers returns a list of the indices of all odd numbers on that list. Sample usage:

```
>>> oddSpots([0,5,7,1,2,5,44,12])
[1, 2, 3, 5]
>>> oddSpots([2,4,8,12])
[]
>>> oddSpots([])
[]
```

3. Write a function `summands` that accepts three arguments, a target number and two lists. The function then returns each pair of numbers, one from the first list and the other from the second list, that sum to the target number. Sample usage:

```
>>> summands( 5, [1,2,3],[1,2,3,5])
[(2, 3), (3, 2)]
>>> summands( 6, [1,2,3],[1,2,3,5])
[(1, 5), (3, 3)]
>>> summands( 12, [1,2,3],[1,2,3,5])
[]
>>> summands( 25, range(20), range(6) )
[]
>>> summands( 25, range(20), range(8) )
[(18, 7), (19, 6)]
```

4. Write a function `tri` that accepts a number n as an argument. The function then **prints** a triangular pattern, see the following samples:

```
>>> tri(3)
321
 21
  1
>>> tri(4)
4321
 321
  21
   1

>>> tri(5)
54321
 4321
  321
   21
    1

>>> tri(7)
7654321
 654321
  54321
   4321
    321
     21
      1
```