

CSC241 – Lab 8

Instructions: Submit a single file lab8.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 `findMinRow()` that takes a two-dimensional list of numbers as a parameter. It **returns** the index of the minimum row in the list. The minimum row is the row with the smallest sum of elements. If the list has multiple rows that achieve the minimum value, the first such row should be returned. If the list is empty the function should return -1. The following shows several sample runs of the function:

```
>>> findMinRow([])
-1
>>> findMinRow( [ [1,2,5], [2,2], [1,3], [7] ] )
1
>>> findMinRow( [ [1,2,5], [2,2], [1,3], [7], [99,99,-200] ] )
4
>>>
```

2. Two words are anagrams if they are the same after rearranging the letters. Write a function `anagram()` that accepts two strings and returns True if they are anagrams and False if not. You may assume that the given arguments consist of only lower case letters. Hint: this is not hard if you use a list. Sample usage:

```
>>> anagram('otto', 'toto')
True
>>> anagram('to', 'otto')
False
>>> anagram('dormitory', 'dirtyroom')
True
>>>
```

3. The mirror of the string 'vow' is the string 'wov' and the mirror image of 'wood' is 'boow'. The mirror of the string 'bed' cannot be represented by a string because the letter 'e' does not have a mirror in the alphabet. Write a function `mirror()` that takes a string and, if it has a mirror, returns its mirror image. If it does not have a mirror, return the string 'INVALID'. You can see how to mirror letters by looking at the last example below:

```
>>> mirror('vow')
'wov'
>>> mirror('wood')
'boow'
>>> mirror('bed')
'INVALID'
>>> mirror('pout')
'tuoq'
>>> mirror( mirror('pout') )
'pout'
>>> [(c,mirror(c)) for c in 'abcdefghijklmnopqrstuvwxyz']
[('a', 'INVALID'), ('b', 'd'), ('c', 'INVALID'), ('d', 'b'),
('e', 'INVALID'), ('f', 'INVALID'), ('g', 'INVALID'), ('h',
'INVALID'), ('i', 'i'), ('j', 'INVALID'), ('k', 'INVALID'),
('l', 'l'), ('m', 'm'), ('n', 'n'), ('o', 'o'), ('p', 'q'),
('q', 'p'), ('r', 'INVALID'), ('s', 'INVALID'), ('t', 't'),
('u', 'u'), ('v', 'v'), ('w', 'w'), ('x', 'x'), ('y',
'INVALID'), ('z', 'INVALID')]
>>>
```

4. Consider this simulation: a random number between 0 and 9 (inclusive of both) is chosen repeatedly until some value occurs more than once. Write a function `numPicks()` that conducts the simulation and returns the count of numbers chosen altogether.

```
>>> random.seed(15)
>>> numPicks() # simulation: 3 0 8 0
4
>>> random.seed(11)
>>> numPicks() # simulation: 7 8 7
3
>>> random.seed(0)
>>> numPicks() # simulation: 6 6
2
```