Python with Applications, PIC16
E. Ryu
Spring 2019

UCLA

Homework 1
Due 5pm, Wednesday, April 10, 2019

Name your file `hw1.py` and submit on CCLE. Comment your code adequately.

Problem 1: Write a function `duplicates(lst)` that `returns` a list of all elements appearing twice or more in the input list `lst`.

Remark. Your function must work with lists that contain lists. For example

```
print duplicates( [ ['a','b'], ['a','b'], 4] )
```

should return `[['a', 'b']]`

Hint. In Python, `sets` cannot contain mutable objects such at `lists`.

Problem 2: Write a function `primeUpTo(n)` that returns a list of all the prime numbers (strictly) less than the input number `n`.

Problem 3: Write a function `longestpath(dict)` that finds the length of a longest path, $(a : b) \to (b : c) \to \cdots$ in a dictionary `dict`. It counts each pointer from a key to a value as one step. For example, the path $(a : b) \to (b : c)$ has length 2. To avoid cycles, we do not allow any key to appear more than once in a path (as a key). Assume `None` does not appear as a key or a value.

Problem 4: Consider the game Hangman. Write the function `hangman(candidate)` that takes in an incomplete word like `p_ck` and `returns` a list of all possible matches like `pack`, `peck`, `pick`, or `puck`. The matches should be words in the dictionary and are case insensitive. The input `candidate` can have 0, 1, 2, or more underscores. Download the file `words.txt` and use the code

```
f = open("words.txt")
word_list = f.read().splitlines()
```

to load the file.

Hint. If you first write a function `match(candidate,word)` `returning` `True` or `False`, then you can implement `hangman(candidate)` in one line using a list comprehension.