

1 Review Exercises

A study guide for exam 3 is available on the course website. This handout presents some review exercises.

1. Write a function that accepts a file name as a parameter (a string), and prints each line of the file in order of longest line to shortest line. The function does not need to return anything.
2. Write a function `pruned_list` that takes a string `frag` a list of strings `words` and returns a new list that contains all words that start with `frag`.

```
>>> L = ['cat', 'horse', 'cattail', 'dog']
>>> pruned_list('cat', L)
['cat', 'cattail']
```

3. Same idea as previous question but *modify* the list rather than return a new list. Write a function `prune_list` that takes a string `frag` a list of strings `words` and *modifies* the list, removing all words that do not start with `frag`. The function should return `None`.

```
>>> L = ['cat', 'horse', 'cattail', 'dog']
>>> prune_list('cat', L)
>>> print L
['cat', 'cattail']
```

4. Al and Buffy go out birding. Each returns with a list of bird sightings. For example, Al returns with `['hawk', 'jay', 'hawk', 'cardinal', 'jay', 'jay']` and Buffy returns with `['kestrel', 'kestrel', 'jay']`. Write a function `eagle_eye_al` that takes in two lists, Al's and Buffy's, and returns the kind that Al saw the most frequently but Buffy did not see it at all.

In the above example, the correct answer is `'hawk'` because Al saw two hawks whereas Buffy saw none. (Al saw jays more frequently than hawks, but Buffy saw a jay so jays are excluded from consideration.)

5. Write a function called `reverseDict` that takes a dictionary as a parameter and returns a new dictionary in which the keys and values of the original dictionary are inverted. In particular, if `k:v` is an entry in the original dictionary, then `v` should be a key in the new dictionary with `k` as a corresponding value.

Because dictionary values are not necessarily unique, values in the new (inverted) dictionary will need to be a list. The list assigned to key `x` in the new dictionary should contain all the keys from the original dictionary for which the value is `x`.

For example,

```
reverseDict({'apple':3, 'potato':2, 'avocado':3})
```

should return the dictionary

```
{2:['potato'], 3:['apple', 'avocado']}
```

6. Consider these two programs.

(a) What is printed?

```
def f(t):
    t = t.upper()
    return t

s = 'abc'
f(s)
print s
```

(b) What is printed?

```
def g(L):
    for i in range(len(L)):
        L[i] = L[i].upper()
    return L

a_list = ['abc', 'xyz']
g(a_list)
print a_list
```

(c) Explain the difference between the two functions in terms of aliasing.

7. For each of the following segments of code, show what the output will be. There are no errors in either code segment.

(a) What is printed?

```
y = ['f', 'x']

def woof(y):
    y[0] = y[-2]
    x = y[0:2]
    x[-1] = '?'
    print "X =", x
    print "Y =", y
    return y
    return x

x = ['a', 'r', 'f']
woof(x)
print "x=", x
print "y=", y
```

(b) What is printed?

```
def bark(x, y):
    print "len(X)=", len(x)
    print "Y=", y
    x['a'].append(3)
    del x['b']
    y.append('!!!')

w = [1,2]
y = {'a': w, 'b': [2,3]}
bark(y, w)
print 'w=' , w
print 'y=' , y
```