CSE 115

Introduction to Computer Science I
Help us help you!

When posting to Piazza:
Help us help you!

When posting to Piazza:

tell us what you think the problem is
Help us help you!

When posting to Piazza:

tell us what you think the problem is

tell us what you've tried
Help us help you!

When posting to Piazza:

tell us what you think the problem is

tell us what you've tried

tell us where you're getting stuck
Help us help you!

When posting to Piazza:

tell us what you think the problem is

tell us what you've tried

tell us where you're getting stuck

Just posting a screenshot of code and saying 'help' isn't an effective learning strategy.
Road map

▶ Review ◀

exercises

Writing files

exercises
1. Define a function that takes a file name as an argument and returns a map with character counts for the file.

def countCharacters(filename):
    count = {}
    with open(filename) as f:
        for line in f:
            for ch in line:
                if ch in count:
                    count[ch] = count[ch] + 1
                else:
                    count[ch] = 1
    return count
Exercises

2. Define a function that takes a file name as an argument and returns a map with word counts for the file.

```python
import re

def countWords(filename):
    count = {}
    with open(filename) as f:
        for line in f:
            wordList = re.split("[^a-zA-Z']+", line)
            for word in wordList:
                if word in count:
                    count[word] = count[word] + 1
                else:
                    count[word] = 1
    return count
```
A csv file can be read from and written to by different applications, such as Excel (left) and Numbers (right).

Heating.csv

```
Month, Budget, Actual
January, 200, 190
February, 200, 210
March, 150, 185
April, 100, 110
May, 50, 40
June, 50, 15
July, 50, 12
August, 50, 14
September, 50, 35
October, 100, 78
November, 150, 125
December, 200, 167
```
import csv

def readBudget(filename):
    budget = {}
    with open(filename, newline='') as f:
        reader = csv.reader(f)
        for line in reader:
            month = line[0]
            line.pop(0)
            budget[month] = line
    return budget
Road map

Review

▶ exercises ◀

Writing files

exercises
Exercise

1. Define a function 'overspent' which takes a dictionary, like the one the readBudget function produces, and returns a dictionary of the months in which expenditures were over the budget, along with the difference (as a negative value).

Remember – the dictionary produces looks like this:

```python
{'Month': ['Budget', 'Actual'], 'January': ['200', '190'], 'February': ['200', '210'], 'March': ['150', '185'], 'April': ['100', '110'], 'May': ['50', '40'], 'June': ['50', '15'], 'July': ['50', '12'], 'August': ['50', '14'], 'September': ['50', '35'], 'October': ['100', '78'], 'November': ['150', '125'], 'December': ['200', '167']}
```
Exercise

1. Define a function 'overspent' which takes a dictionary, like the one the readBudget function produces, and returns a dictionary of the months in which expenditures were over the budget, along with the difference (as a negative value).

```python
def overspent(budget):
    over = {}
    months = list(budget.keys())
    months.pop(0)
    for key in months:
        data = budget[key]
        diff = int(data[0]) - int(data[1])
        if (diff < 0):
            over[key] = diff
    return over
```
Exercise

2. Define a function 'underspent' which takes a dictionary, like the one the readBudget function produces, and returns a dictionary of the months in which expenditures were under budget, along with the difference (as a positive value).

```python
def underspent(budget):
    under = {}
    months = list(budget.keys())
    months.pop(0)
    for key in months:
        data = budget[key]
        diff = int(data[0]) - int(data[1])
        if diff > 0:
            under[key] = diff
    return under
```
Road map

Review

exercises

▶ Writing files ◄

exercises
def write(filename, contents):
    with open(filename, 'w') as f:
        for item in contents:
            f.write(item + '\n')
def write(filename, contents):
    with open(filename, 'w') as f:
        for item in contents:
            f.write(item + '
')

The write function expects a string.
To print other types of values, first convert them to an equivalent string.

For example:
    f.write(str(7))
import csv

def writeCSV(filename, dataTable):
    with open(filename, 'w', newline='') as f:
        writer = csv.writer(f)
        for record in dataTable:
            writer.writerow(record)
import csv

def writeCSV(filename, dataTable):
    with open(filename, 'w', newline='') as f:
        writer = csv.writer(f)
        for record in dataTable:
            writer.writerow(record)

This writes the members of record on one line, separated by commas.
dt = [['abc', 'def'], ['ghij', 'klmn']]
writeCSV('file1', dt)

writeCSV('file2', dt[0])

Writing csv files
Exercise

Define a function which, given the name of a csv file, reads data from that csv file. Each record in the file has fields $f_0$ through $f_N$. Write a new csv file, whose file name is the same as the original but prefixed with 'R', which as the same records but with the fields reversed, from $f_N$ through $f_0$. 