

PPiC 5.6 Write a program that reads in a file and then prints out the number of lines, words, and characters in the file.

```
>>> def read_print(file):
    fref = open(file, 'r')
    countl = 0
    countw = 0
    chnum = 0
    for l in fref:
        countl = countl + 1
        words = l.split()
        countw = len(words) + countw
    print(file, 'has', countl, 'lines;')
    print(file, 'has', countw, 'words;')
    fref.close()
    fref = open(file, 'r')
    for ch in fref.read():
        characters = len(ch)
        chnum = characters + chnum
    num = chnum - countl*2
    print(file, 'has', num, 'characters.')
    fref.close()
```

PPiC 5.7 Write a program that creates a file with a concordance--an index that tells you which line of the file each word appears on. If a word is on more than one line, the concordance will show you all of the lines containing that word. *Hint:* Use a dictionary keyed by each word to solve this problem.

```
>>> def concordance(file):
    fref = open(file, 'r')
    line = fref.readline()
    Dict = {}
    lnum = 0
    while line is not '':
        lnum = lnum + 1
        words = line.split()
        for word in words:
            word = word.replace(',', '')
            if word not in Dict:
                Dict[word] = [lnum]
            elif Dict[word][-1] != lnum:
                Dict[word].append(lnum)
        line = fref.readline()
    return Dict
```

```
>>> concordance("rainfall.txt")
{'33.95': [7], 'AmesW': [6], '33.64': [4], '35.27': [12], '38.02': [17], 'Beaconsfield': [12], '34.07': [6], '33.59': [19], '33.41': [11], 'BellePlaine': [14], 'Ankeny': [9], 'Blockton': [16], '36.35': [13], '33.33': [24], '25.81': [1], '27.43': [5], 'Carroll': [24], 'AmesSE': [7], 'Cascade': [25], '33.48': [25], 'Alton': [5], 'Britt': [20], 'Akron': [1], 'Anamosa': [8], 'Allison': [4], '36.28': [16], 'Algona': [3], 'Bloomfield': [17], '36.30': [18], '34.35': [15], '31.54': [20], 'Burlington': [23], 'Bedford': [13], 'BurlingtonKBUR': [22], 'Buckeye': [21], 'Brighton': [19], 'Audubon': [11], '37.65': [2], '33.66': [21], 'Atlantic': [10], '35.33': [8], '30.69': [3], '37.94': [22], 'Boone': [18], '33.38': [9], '34.77': [10], 'Albia': [2], 'Bellevue': [15], '36.94': [23], '35.81': [14]}
```

PPiC 5.13 Write a function called `savePage` that takes a string representing a URL, and a file name as a parameter and then saves the contents of the web page to the file.

```
>>> def savePage(url, file):
    import urllib.request
    page = urllib.request.urlopen(url)
    pageText = page.read()
    decodedPageText = pageText.decode('utf-8')
    page.close()
    fref = open(file, 'w')
    fref.write(decodedPageText)
    fref.close()
    return fref

>>> url="https://raw.githubusercontent.com/ncoop/python-context/master/resources/cs150exams.txt"
>>> file = "rainfall.txt"
>>> savePage(url, file)
<_io.TextIOWrapper name='rainfall.txt' mode='w' encoding='US-ASCII'>
>>>
```

PPiC 5.18 Use a *while* loop to implement the *for* loop for `i in range(10)`.

```
i = 0
while i <= 9:
    i = i + 1
```