MSDS 7330
File Organization and Database Management

Mini Project 5
XML

Evangelos Giakoumakis

Question 1: The file baseball salaries 2003.txt contains salary information for certain professional baseball players from the year 2003. Define an XML schema for this file. Write a Python script that processes this file and stores it in a single XML file.

XML Schema:

```
<!DOCTYPE baseball [
<!ELEMENT Team ( #PCDATA )>
<!ELEMENT Player ( #PCDATA )>
<!ELEMENT Salary ( #PCDATA )>
<!ELEMENT Position ( #PCDATA )>
] >
```

Python Script:

```
# xmlparser.py
# First row of the csv file must be header!

# example CSV file: baseball_salaries_2003.csv
# team,player,salary,position
# New York Yankees ,"Acevedo Juan",900000,Pitcher

import csv

csvFile = 'baseball_salaries_2003.csv'
xmlFile = 'baseball_salaries_2003.xml'

csvData = csv.reader(open(csvFile))
xmlData = open(xmlFile, 'w')
xmlData.write('<?xml version="1.0"?>' + "\n")
# there must be only one top-level tag
xmlData.write('<csv_data>' + "\n")

rowNum = 0
for row in csvData:
   if rowNum == 0:
      tags = row
      # replace spaces w/ underscores in tag names
      for i in range(len(tags)):
         tags[i] = tags[i].replace(' ', '_')
   else:
      xmlData.write('<row>' + "\n")
      for i in range(len(tags)):
         xmlData.write('    ' + '<' + tags[i] + '>' \
                    + row[i] + '</' + tags[i] + '>' + "\n")
      xmlData.write('</row>' + "\n")

   rowNum +=1

xmlData.write('</csv_data>' + "\n")
xmlData.close()
```

Baseball salaries 2003.xml (first 3 rows)

```
<?xml version="1.0"?>
<csv_data>
<row>
   <Team>New York Yankees </Team>
   <Player>Acevedo Juan  </Player>
   <Salary>900000</Salary>
   <Position> Pitcher</Position>
</row>
<row>
   <Team>New York Yankees </Team>
   <Player>Anderson Jason</Player>
   <Salary>300000</Salary>
   <Position> Pitcher</Position>
</row>
<row>
   <Team>New York Yankees </Team>
   <Player>Clemens Roger </Player>
   <Salary>10100000</Salary>
   <Position> Pitcher</Position>
</row>
```

Question 2: The file baseball salaries 2003.xml contains salary information for certain professional baseball players from the year 2003. Write a Python script that processes the XML file from Question 1 to determine, for each position, the average salary of the players in that position. Note that the seven player positions that can occur in the input file are “Catcher”, “First Baseman”, “Outfielder”, “Pitcher”, “Second Baseman”, “Shortstop” and “Third Baseman”. The output should appear sorted in descending order of average salary.

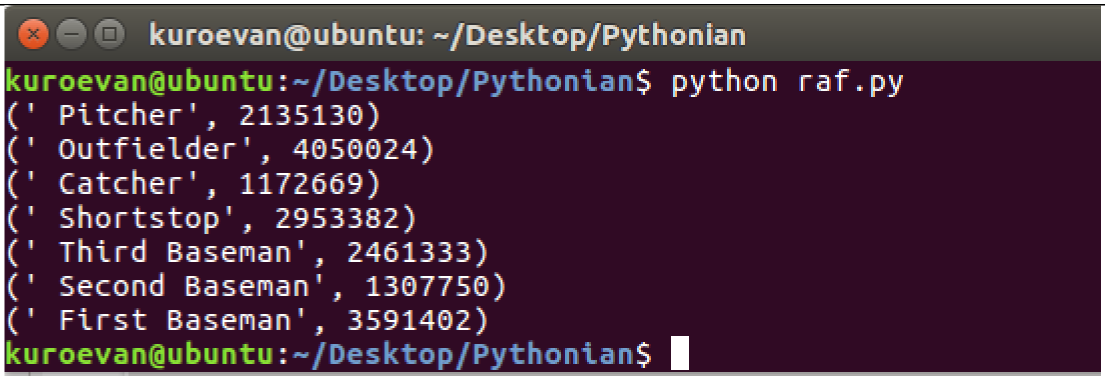
Python XML processing script:

```
from xml.etree import ElementTree

outfile = "output.txt"
filename = "baseball_salaries_2003.xml"
dom = ElementTree.parse(filename)

# Pitcher

count = 0
total = 0
avg = 0

rows = dom.findall('row')

for c in rows:
    pos = c.find('Position').text
    sal = c.find('Salary').text

    if pos == ' Pitcher':
        count+=1
        total = total + int(sal)

avg = total / count
print(pos, avg)

# Outfielder

count2 = 0
total2 = 0
avg2 = 0

rows = dom.findall('row')

for d in rows:
    pos = d.find('Position').text
    sal = d.find('Salary').text

    if pos == ' Outfielder':
        count2+=1
        total2 = total2 + int(sal)

avg2 = total2 / count2
print(' Outfielder', avg2)

# Catcher
```

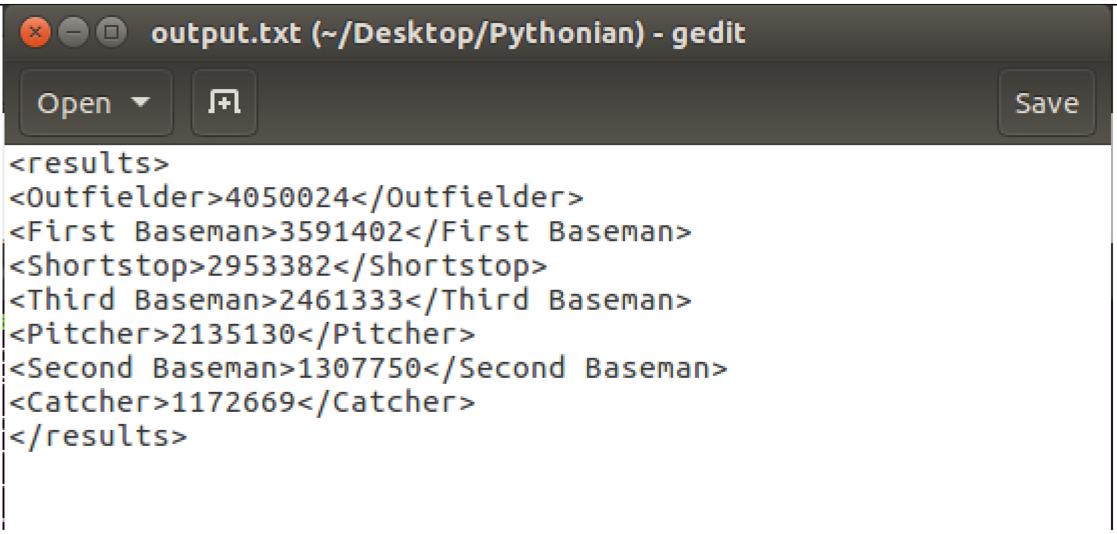
```
count3 = 0
total3 = 0
avg3 = 0

rows = dom.findall('row')

for e in rows:
    pos = e.find('Position').text
    sal = e.find('Salary').text

    if pos == ' Catcher':
        count3+=1
        total3 = total3 + int(sal)

avg3 = total3 / count3
print(' Catcher', avg3)

# Shortstop

count4 = 0
total4 = 0
avg4 = 0

rows = dom.findall('row')

for c in rows:
    pos = c.find('Position').text
    sal = c.find('Salary').text

    if pos == ' Shortstop':
        count4+=1
        total4 = total4 + int(sal)

avg4 = total4 / count4
print(' Shortstop', avg4)

# Third Baseman

count5 = 0
total5 = 0
avg5 = 0

rows = dom.findall('row')

for c in rows:
    pos = c.find('Position').text
    sal = c.find('Salary').text
    if pos == ' Third Baseman':
        count5+=1
        total5 = total5 + int(sal)

avg5 = total5 / count5
print(' Third Baseman', avg5)

# Second Baseman

count6 = 0
total6 = 0
avg6 = 0

rows = dom.findall('row')

for c in rows:
    pos = c.find('Position').text
    sal = c.find('Salary').text

    if pos == ' Second Baseman':
        count6+=1
        total6 = total6 + int(sal)

avg6 = total6 / count6
print(' Second Baseman', avg6)

# First Baseman

count7 = 0
total7 = 0
avg7 = 0

rows = dom.findall('row')

for c in rows:
    pos = c.find('Position').text
    sal = c.find('Salary').text

    if pos == ' First Baseman':
        count7+=1
        total7 = total7 + int(sal)

avg7 = total7 / count7
print(' First Baseman', avg7)

# print to file

from xml.etree.ElementTree import Element, SubElement, tostring
```

```
res = Element('results')

child = SubElement(res, 'Outfielder')
child.text = str(avg2)

child = SubElement(res, 'First Baseman')
child.text = str(avg7)

child = SubElement(res, 'Shortstop')
child.text = str(avg4)

child = SubElement(res, 'Third Baseman')
child.text = str(avg5)

child = SubElement(res, 'Pitcher')
child.text = str(avg)

child = SubElement(res, 'Second Baseman')
child.text = str(avg6)

child = SubElement(res, 'Catcher')
child.text = str(avg3)

f = open(outfile,'w')

print >>f, tostring(res)
```

Output:



```
kuroevan@ubuntu:~/Desktop/Pythonian$ python raf.py
(' Pitcher', 2135130)
(' Outfielder', 4050024)
(' Catcher', 1172669)
(' Shortstop', 2953382)
(' Third Baseman', 2461333)
(' Second Baseman', 1307750)
(' First Baseman', 3591402)
kuroevan@ubuntu:~/Desktop/Pythonian$
```



```
<results>
<Outfielder>4050024</Outfielder>
<First Baseman>3591402</First Baseman>
<Shortstop>2953382</Shortstop>
<Third Baseman>2461333</Third Baseman>
<Pitcher>2135130</Pitcher>
<Second Baseman>1307750</Second Baseman>
<Catcher>1172669</Catcher>
</results>
```

Note: Please find attached files alongside document.