Numerical Simulation

Arrays & Plotting
Coursework
exam2 this week, 2/28–3/2
- Study HPL Qs 2, 6.1–10
- hw04 due today, 2/27
- hw05 will be assigned next week
Recap: Dictionaries
Question #1

x = 'ABBACAB'
d = {}
for c in x:
    if c not in d:
        d[c] = 0
    d[c] += 1

What is the final value of d?
A { 'A': 3, 'C': 1, 'B': 3 }
B { 'A': 3, 'B': 3, 'C': 1 }
C { 'AAA', 'BBB', 'C' }
D [ 3, 3, 1 ]
x = 'ABBACAB'
d = {}
for c in 'ABBACAB':  # c = 'A', 'B', etc.
    if c not in d:
        d[c] = 0
        d[c] += 1
c='A' => d['A']=0
c='A' => d['A']+=1
c='B' => d['B']=0
c='B' => d['B']+=1
...
x = 'ABBACAB'
d = {
    for c in x:
        if c not in d:
            d[c] = 0
        d[c] += 1

What is the final value of d?
A { 'A': 3, 'C': 1, 'B': 3 } ★
B { 'A': 3, 'B': 3, 'C': 1 } ★
C { 'AAA', 'BBB', 'C' }
D [ 3, 3, 1 ]

Why both?
d = {}
for n in d:
    print( n )

What does this code print?
A  The values of d.
B  The keys of n.
C  The key-value pairs of d.
d = { 'red':1, 'green':2, 'blue':3 }
for n in d:
    print( n )

red
blue
green
d = {}
for n in d:
    print( n )

What does this code print?
A  The values of d.
B  The keys of n.  ⋆
C  The key-value pairs of d.

So how do you access value corresponding to key n?  d[ n ]
myfile = open( 'odyssey.txt' )
text = myfile.read()
d = {}
for l in text.split():
    if l not in d:
        d[ l ] = 0
    d[ l ] += 1

What does this code do?
   A Counts all of the lines in 'odyssey.txt'.
   B Counts all of the words in 'odyssey.txt'.
   C Counts all of the characters in 'odyssey.txt'.
myfile = open('odyssey.txt')
text = myfile.read()
d = {}
for l in text.split():
    if l not in d:
        d[l] = 0
    d[l] += 1

- What is type of text?
- What is type of text.split()?  
- What is type of l?
- What is contents of d?
  d['Tell'] == 30
d['tell'] == 227
myfile = open( 'odyssey.txt' )
text = myfile.read()
d = {}
for l in text.split():
    if l not in d:
        d[ l ] = 0
    d[ l ] += 1

What does this code do?
A  Counts all of the lines in 'odyssey.txt'.
B  Counts all of the words in 'odyssey.txt'. ⋆
C  Counts all of the characters in 'odyssey.txt'.
zipcode = { 'Bill': 60644,
           'Jill': 41073,
           'Tony': 63103 }
city = { 60644: 'Chicago',
          41073: 'Cincinnati',
          63103: 'St. Louis' }
x = city[ zipcode[ 'Tony' ] ]

What is the final value of x?
A '63103'
B 'St. Louis'
C 'Chicago'
D None
E I’m lost.
zipcode = { 'Bill': 60644,
           'Jill': 41073,
           'Tony': 63103 }
city = { 60644: 'Chicago',
         41073: 'Cincinnati',
         63103: 'St. Louis'}
x = city[ zipcode['Tony'] ]

What is the final value of x?
A '63103'
B 'St. Louis' *
C 'Chicago'
D None
E I’m lost.
Arrays
mydata = [ 4.5, 6.0, 1.2, 5.4 ]
from math import sin
sin(mydata)

- Why doesn’t this work?
- list can contain any type!
- Also operators don’t do what we “want”:
  mydata * 2.0  # doesn’t double values!
import numpy
import numpy as np  # better way

- numpy provides arrays and mathematical functions.

data = np.array([4.5, 6.0, 1.2, 5.4])
data * 2.0
numpy

```python
x = np.array([[1,2], [3,4]])
np.zeros((3,3))
np.ones((4,4))
np.eye(4)
x.shape
x.dtype
x = np.array([[1,2], [3,4]], dtype=np.float64)
x[:,1]
x[1, :]
x.T
x.tolist()
x.sort()
x.argsort()
```
Consider a data set containing patient inflammation records for 60 patients over a period of 40 days, contained in inflammation.csv.

```python
data = np.loadtxt('./data/inflammation.csv', delimiter='','
```
Max for each patient
`data.max(axis=1)`

Daily Avg
Average for each day
`data.mean(axis=0)`
Plotting
import matplotlib.pyplot as plt
%matplotlib inline  # jupyter only

- A plotting environment similar to MATLAB.
- Can plot lists or arrays.

xs = list( range(4) )
ys = [ 4.5, 6.0, 1.2, 5.4 ]
plt.plot( xs, ys )
plt.show()
Always include labels:

```python
plt.xlabel( 'domain' )
plt.ylabel( 'range' )
plt.title( 'topical data' )
plt.plot( xs, ys )
plt.xlabel( 'x' )
plt.ylabel( 'y' )
plt.title( 'some values' )
plt.show()
```
Basic cycle:
- Add data to plot.
- Add labels to plot.
- Show plot.
Two kinds of plots today:

- `plt.plot(x, y) # for ptwise data`
- `plt.imshow(A) # for array data`

`plot`: third argument is *format string* (optional).

```python
plt.plot(x, y, 'r.')
plt.show()
```

`plot`: can also take keyword arguments.

```python
plt.plot(x, y, 'r.', label='height')
plt.show()
```
Next steps
Next steps

- exam2 this week, 2/28–3/2
- hw04 due today, 2/27
- hw05 will be assigned next week
- Read for the next class