Lab Final Reference Notes

Python

```python
import numpy #import the numpy module
from math import sqrt, log10 # useful functions

b = numpy.zeros(n, dtype='float') # initialize a n x 1 vector of zeros
A = numpy.zeros([n, m], dtype='float') # initialize a n x m array of zeros
c = numpy.dot(A, b) # dot product of two arrays

if condition1:
    # do something
elif condition2:
    # do something else
else:
    # do something else

for i in range(start, stop, step):
    # do something

i = 1
while i < 10:
    # do something
    i = i + 1

def myfun(x1,...,xM):
    # do something
    return y1, ..., yN

A.shape # size of array

# solve Ax=b for x
x = np.linalg.solve(A, b)

# plot 2 datasets on a graph
import matplotlib.pyplot as plt # module for plotting
plt.plot(x, y, label='dataset1')
plt.plot(x2 y2 label='dataset1')
plt.legend()
plt.show()

# print x
print("The value of x is = ", x)
```

Matlab Reference

```
b = zeros(n, 1) % initialize a n x 1 vector of zeros
A = zeros(n) % initialize a n x m array of zeros
c = A * b % dot product of A and b

if condition1
    % do something
else if condition2
    % do something else
else
    % do something else
end

for i=start:step:stop
    % do something

i = 1
while(i < 10)
    % do something
    i = i + 1

function [y1,...,yN] = myfun(x1,...,xM)
    % do something

size(A) % size of array

# solve Ax=b for x
x = linsolve(A, b)
# or, alternatively
x = A/b

% plot 2 datasets on a graph
plot(x1,y1)
hold on
plot(x2,y2)
legend('dataset1','dataset2')
plot(x, y)
```