Pre-Lab MatLab

1. a=zeros(1,5)
   1. output: a= 0 0 0 0 0
   2. creates a 1 by 5 array with each entry being a zero
2. b=[3:1:5]
   1. output: b = 3 4 5
   2. creates an array starting from 3 ending at 5 with an increment of one
3. c=[1:-3.5:-9]
   1. output: c = 1.0000 -2.5000 -6.0000
   2. creates an array starting at 1 and ending at -9 with an increment of -3.5. However it does not reach -9 and stops at -6 because it cannot go to -9.5 because it is outside the range
4. d = [1.4, 2.3; 5.1, 7.8]
   1. output: d= 1.4000 2.3000
   2. 5.1000 7.8000
   3. this creates a 2 by 2 array with the given input values
5. e=d(1,2)
   1. output: e = 2.3000
   2. gets the element from the first row second column
6. f=d(:,2)
   1. output f= 2.3000
   2. 7.8000
   3. access the second column and transposes it to a vertical matrix.
7. d.^2
   1. output: d = 1.9600 5.2900
   2. 26.0100 60.8400
   3. raises each element to the second power
8. comp = 3+4i
   1. output: 3.0000 + 4.0000i
   2. assign 3+4i to the variable comp
9. real(comp)
   1. output: 3
   2. returns the real part of 3.0000 + 4.0000i
10. abs(comp)
    1. ouput: 5
    2. takes the absolute value of 3 + 4i, since it is a complex number the return value is sqrt((3^2)+(4^2))
11. 3 ~= 5
    1. output: 1
    2. the Boolean not equals operator. Since 3 does not equal 5 it returns 1 which in matlab means true

Part B

Pre-Lab: Python

1. a = numpy.zeros(3)
   1. output: array([ 0., 0., 0.])
   2. creates a 1 by 3 matrix with each entry being a zero.
2. b = a.shape
   1. output: (3,)
   2. returns the dimension of the array
3. numpy.absolute([-5.2, -3])
   1. output: array([5.2,3. ])
   2. return the array with each element the absolute value of the input element
   3. the 3. Is there because it convert the -3 to a float
4. c = numpy.arange(5,8,1)
   1. output: array([5,6,7])
   2. the 5 and 8 are the range and the 1 is the step.
   3. This creates an array starting at 5 and goes up to 7, ie one less than 8.
   4. The one is how much to count by. If the one was instead a 2 the output would be array([5,7]) because it is counting by two
5. d = numpy.arange(1,-16.5,-3.5)
   1. output: array([ 1. , -2.5, -6. , -9.5, -13. ])
6. e = numpy.array([[1.4, 2.3], [5.1, 7.8]])
   1. output: array([[ 1.4, 2.3], [ 5.1, 7.8]])
7. f = e[0][0]
   1. output: 1.3999999999999
8. g = e[0][:]
   1. output: array([1.4, 2.3])
9. comp = 5+12j
   1. ouput: (5+12j)
   2. assigns 5+12j to the variable comp
   3. j is the imaginary value sqrt(-1)
10. numpy.imag(comp)
    1. output: array(12.0)
    2. returns the imaginary part of the elements of the array (taken from numpy documentation)
11. numpy.absolute(comp)
    1. output: 13.0
    2. returns the absolute value of the input
    3. because comp is imaginary it returns sqrt((5^2)+(12^2))
12. 5 == 5
    1. output: True
    2. this is checks if 5 is equal to 5 this is true so the output is true
    3. this is a Boolean expresion

Part B.

1. 
2. 
3. 

Part C

import numpy

def clip(x):

y = x

for i in numpy.nditer(y, op\_flags=['readwrite']):

if i[...] < 5:

i[...] = 5

return y

x = numpy.array([3,8,6,2,7])

clip = clip(x)

print clip

output is [5 8 6 5 7]