Problem 1:

```
import perc
import matplotlib.pyplot as plt

def plot(l, p):

        grid = perc.percolation2d('biggest', l, p)

        plt.imshow(grid, cmap = "cool", interpolation = "nearest")

        cax = plt.imshow(grid, interpolation='nearest', cmap='cool')

        cbar = plt.colorbar(cax, ticks=[0, 1, 2])
        cbar.ax.set_yticklabels(['No Cluster', 'Small Clusters', 'Largest Cluster'])

        plt.xlabel('X Position')
        plt.ylabel('Y Position')
        plt.title('Largest Cluster for p = %s' % p)

plot(1000, .5)
```

Problem 2:

```
import perc
import matplotlib.pyplot as plt
import numpy as np


def percent_largest(d, lmax):

        p = .01
        q = []

        while p <= 1:
        size = 0
        total = 0

        for i in xrange(10):
        size += perc.biggest_clusters(d, lmax, p, 1)[0][2][0]
        total += ((lmax**2) * p)

        size1 = size/10.0
```

```
        total1 = total/10.0

        p += .01

        q.append(size1/total1)

        return q

def percent_second_largest(d, lmax):

        p = .01
        q = []

        while p <= 1:
        size = 0
        total = 0

        for i in xrange(10):
        size += perc.biggest_clusters(d, lmax, p, 2)[0][2][1]
        total += ((lmax**2) * p)

        size1 = size/10.0
        total1 = total/10.0

        p += .01

        q.append(size1/total1)

        return q

def plot(d, l):

        a = percent_largest(d, l)
        b = percent_second_largest(d, l)
        x = np.arange(.01, 1, .01)

        plt.plot(x, a, label = 'q_1')
        plt.plot(x, b, label = 'q_2')
        plt.legend()
        plt.xlabel('Probability p')
        plt.ylabel('Log of Probability q')
        plt.yscale('log')
```

```
        plt.title('Probability a Randomly Chosen Point is Part of the Two Largest Clusters in
%sD' % d)

plot(3, 30)
```

Problem 3:

```
import perc
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit


def largest_clusters(d, pc, lmax):

        length = range(2, lmax)
        size = np.array([0]*len(length))

        for i in xrange(10):
        clusters = perc.biggest_clusters(d, range(2, lmax), pc, 2)
        size += np.array([clusters[i][2][0] for i in xrange(len(clusters))])

        size = size/10.0

        return size


def plot_pc(d, pc, lmax):

        x = np.log(xrange(2, lmax))
        y = np.log(largest_clusters(d, pc, lmax))

        def func(x, a, b): #generates function to be fit against
        return a*x + b

        popt, pcov = curve_fit(func, x, y)

        print "a = %s , b = %s" % (popt[0], popt[1])

        plt.plot(x, func(x, *popt), label="Fitted Curve")

        plt.plot(x, y, label = 'Simulated Data (%s * x + (%s))'% (round(popt[0], 2),
round(popt[1], 2)))
```

```
        plt.legend()
        plt.xlabel('Log of Grid Side Length L')
        plt.ylabel('Log of Size of Two Largest Clusters')
        plt.title('Size of Largest Clusters in 2D as a Function of Grid Size p = %s' % pc)


def plot_p_less(d, pc, lmax):

        x = np.log(xrange(2, lmax))
        y = largest_clusters(d, pc, lmax)

        def func(x, a, b): #generates function to be fit against
        return a*x + b

        popt, pcov = curve_fit(func, x, y)

        print "a = %s , b = %s" % (popt[0], popt[1])

        plt.plot(x, func(x, *popt), label="Fitted Curve")

        plt.plot(x, y, label = 'Simulated Data (%s * x + (%s))'% (round(popt[0], 2),
round(popt[1], 3)))
        plt.legend()
        plt.xlabel('Log of Grid Side Length L')
        plt.ylabel('Size of Two Largest Clusters')
        plt.title('Size of Second Largest Clusters in 2D as a Function of Grid Size p = %s' % pc)


def plot_p_more(d, pc, lmax):

        x = np.log(xrange(2, lmax))
        y = np.log(largest_clusters(d, pc, lmax))
        print x
        print y

        def func(x, a, b): #generates function to be fit against
        return a*x + b

        popt, pcov = curve_fit(func, x, y)

        print "a = %s , b = %s" % (popt[0], popt[1])

        plt.plot(x, func(x, *popt), label="Fitted Curve")
```

```
        plt.plot(x, y, label = 'Simulated Data (%s * x + (%s))'% (round(popt[0], 2),
round(popt[1], 3)))
        plt.legend()
        plt.xlabel('Log of Grid Side Length L')
        plt.ylabel('Log of Size of Two Largest Clusters')
        plt.title('Size of Largest Clusters in %sD as a Function of Grid Size p = %s' % (d, pc))

plot_p_less(2, .57, 200)
```