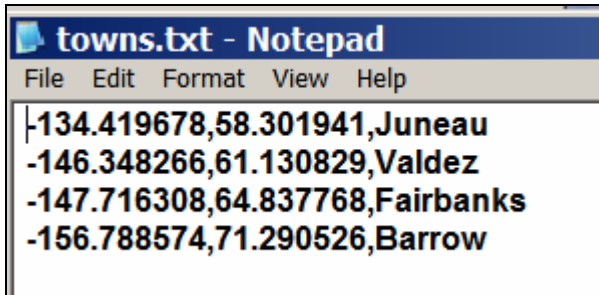


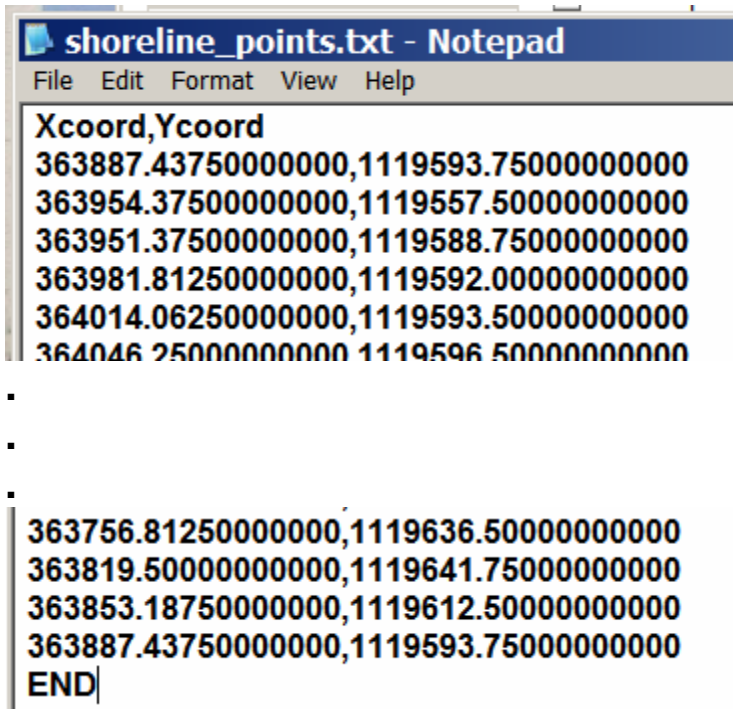
Python Geoprocessing : Creating Feature Classes From Text Files Due by Friday 6pm

Download the textfiles points.txt and polygons.txt from
<http://nrm.salrm.uaf.edu/~dverbyla/nrm638/data>

The textfiles can be used for coordinates to create point and polygon themes.



```
towns.txt - Notepad
File Edit Format View Help
-134.419678,58.301941,Juneau
-146.348266,61.130829,Valdez
-147.716308,64.837768,Fairbanks
-156.788574,71.290526,Barrow
```



```
shoreline_points.txt - Notepad
File Edit Format View Help
Xcoord,Ycoord
363887.43750000000,1119593.75000000000
363954.37500000000,1119557.50000000000
363951.37500000000,1119588.75000000000
363981.81250000000,1119592.00000000000
364014.06250000000,1119593.50000000000
364046.25000000000,1119596.50000000000
.
.
.
363756.81250000000,1119636.50000000000
363819.50000000000,1119641.75000000000
363853.18750000000,1119612.50000000000
363887.43750000000,1119593.75000000000
END
```

Do one of the following:

- 1) Write a python script that will create a point feature class called **towns.shp** from the towns.txt file. Each point should include a town_name field with the correct town (use the **gp.AddField** tool to add the field to your table after you create the feature class). Write the output from any Geoprocessing tools to a log file called **towns.log**.

You can use the find and rfind methods to find the left-most and right-most string position of the comma after your read a line from the text file.

For example:

```
>>> line = file.readline() #read first line from text file
>>> line = line[:-1] #strip off newline character
>>> line
'-134.419678,58.301941,Juneau'
>>> first_comma = line.find(',')
>>> last_comma = line.rfind(',')
>>> first_comma,last_comma #positions in string
(11, 21)
>>> line
'-134.419678,58.301941,Juneau'
>>> x_string = line[0 : first_comma]
>>> y_string = line[first_comma + 1 : last_comma]
>>> name = line[ last_comma + 1 : ]
>>> print x_string,y_string,name
-134.419678 58.301941 Juneau
>>> X= float(x_string)
>>> Y =float(y_string)
```

OR

- Write a python script that will create a polylines feature class called **shoreline.shp** from the shoreline_points.txt file. First read the text file to determine the number of points defining the shoreline of Montegue Island in Prince William Sound. For example:

```
file = open('c:/temp/shoreline_points.txt','r')
line = file.readline() #first line Xcoord, Ycoord
print line
count = 0
while line :
    line = file.readline()
    print line
    count = count + 1
file.close()
print 'Number of x,y coords:',count
```

Start by copying your shoreline_points.txt file to a test.txt file and edit it so it contains only three points. Write your script so it works with this simple file first, then modify your script so that it works with the large file.

Print a status message “___percent converted” to the user as your script is being executed. Loop through the points until the keyword END is read from the input file. Write any Geoprocessing messages to a log file.

Email me (D.Verbyla@uaf.edu) your python script and log file by Friday.

<http://nrm.salrm.uaf.edu/~dverbyla/nrm638/>