**Task 1 (20 marks): Encode and decode some text**

Suppose you have a message in English of a sufficient length, say 100 charachters. The goal of this task will be to turn that text into a list of integers, encode it by applying an operation on the indexes, and then decode it back to reproduce the original message. You can think of this as an exercise in making a message secret by encoding it and then decoding it.

The following string contains all of the printable characters found on most keyboards, so it can be used for turning the message into integers:

all\_char = string.printable

so that letter 'c' corresponds to index 12 in all\_char.

In the cell below, write a piece of code that will do the following work:

* (1) Introduce a message to work with (a string of English text, should be from 50 to 100 charachters).
* (2) Create a list with symbols from your message. For example, the message secret becomes the list [28, 14, 12, 27, 14, 29] in terms of the list all\_char. [marks 2]
* (3) Now encode the message by incrementing the integers in the list by a number, e.g, 10, so that your secret list from above becomes [38 24 22 37 24 39]. The encoding number is a key that can be used to decode it.[marks 1]
* (4) Now add to this list the key as the last element, so that your list becomes in the examle above [38 24 22 37 24 39 10] [marks 1]
* (5) Print the encoded array
* (6) Write a piece of code to decode the message using the key [4]
* (7) Now assume that you do not know that the last element in the array is the key, but you do wish to read the message. You also know how the message was encoded (all integers shifted by some number). Assume that the secret message is likely to contain common English words 'the', 'and', 'have'. Write a piece of code that uses this condition to guess what the value of the key is. [4]
* (8) Print the decoded message.
* (9) A programmer decides to encode the message by raising the indexes to power a, where a is an arbitrary positive number, e.g., 3. Decoding then works by taking the inverse power (b = 1/a) of the array. For example, for a = 3, and the array secret = [28, 14, 12, 27, 14, 29], the first index of the encoded array becomes 28\*\*3 = 21952. The first element of the decoded array is then 21952\*\*b = 28. Try to write a code to do such an encoding and decoding for an arbitrary positive value of a (e.g., a = 2.71). [3]
  + (10) The method may work for some values of a but not for others. Explain what is the problem with the method and how/whether it can be fixed. [3]
  + (11) How would you try to decode the message if you do not know the value of a? [marks 2]

**Task 1 (10 marks): Reading and plotting the data.**

In this task we will work with the file 'cassini\_coords.txt' which was read in and plotted in Section 06. The code below is a copy from that section. To run it, make sure that the data file is in the same section as your test. The data are saved in the structure scdat.

Our goal is to plot rdot not versus r as in the example given in that section but as a function of date expressed as a numeric value. For example, the date 1998-Jan-24 is to be replaced with 1998.0657 because the year contained 365 days and 24/365 = 0.06575. (Hint: remember that some years, like 2000, had 366 days.) Thus,

* (1) Convert the date in the numeric value. Write your own function to do this by decomposing the date into the year, month and the day, and returning the numeric value. [marks 6]
* (2) Plot rdot versus the numeric date for each year. The data for each year should be shown with a different color and a different line style. [marks 4]

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

plt.ion()

datafile = 'cassini\_coords.txt'

scdat = np.recfromtxt(datafile, skip\_header=103, skip\_footer=47,

names=['date', 'time', 'hlon', 'hlat', 'r', 'rdot'])

print scdat.dtype.names

print scdat[0]

print ' '

('date', 'time', 'hlon', 'hlat', 'r', 'rdot')

('1997-Oct-16', '00:00', 22.6208, 0.0137, 0.997922579321, 2.343983)