MAE 8 - Winter 2015
Homework 7

Instructions: Follow the homework solution template. Put all answers in a MATLAB script named **hw7.m**. For this homework, you will need to submit multiple files. Create a zip archive named **hw7.zip**. The zip archive should include the following files: **hw7.m**, **get_daily_T.m**, **get_monthly_T.m**, and **get_yearly_T.m**. Submit **hw7.zip** through TED before 9 PM on 02/26/2015. Use double precision unless otherwise stated.

Problem 1: Create a data structure named **element** to store information about elements in the periodic table. The data structure should have the following fields: name, symbol, atomicNumber, atomicWeight, and electrons. The name and symbol fields are strings. The atomicNumber field is an integer while the atomicWeight is type double. The electrons field is a seven-element vector for the number of electrons in each valence shell.

(a) Fill the structure with the data for the element Helium, which has the following properties: name=Helium, symbol=He, atomicNumber=2, atomicWeight=4.002602, and electrons=[2 0 0 0 0 0 0]. Set **p1a=element**.

(b) Use function **struct2cell** to convert the data structure into a cell array. Put the new cell array into **p1b**.

(c) Was any information lost in the conversion from a data structure to cell array? Your answer should be reported as **p1c='... is lost'**.

(d) Using function **fieldnames**, extract the names of the fields in the data structure **element** to a new variable named **fname**. Set **p1d=fname**.

(e) Use function **cell2struct** to convert the cell array in part (b) back to a structure. Store the structure in **p1e**.

Problem 2: Use a nested structure named **contact** to store the following contact list with the fields: name, phone and email. The phone and email fields are also data structures.

Name	phone		email	
	work	home	work	home
James Smith	619-793-2323	858-516-7812	jsmith@gmail.com	jsmith@yahoo.com
	619-793-3333			
Mary Lee	619-808-7523	858-534-9875	mlee@gmail.com	mlee123@hotmail.com
	619-808-7555			

(a) Get the contact information (all phone numbers and email addresses) for James Smith and store it in **p2a**.

(b) Get the contact information (all phone numbers and email addresses) for Mary Lee and store it in **p2b**.

(c) Remove the second work phone number of James Smith and set **p2c = contact(1).phone.work**.

(d) Remove the first work phone number of Mary Lee and set **p2d = contact(2).phone.work(1)**.

Problem 3: In this exercise, you will look at a database of daily temperature of San Diego and Rapid City from 1995 to 2010. Download the file **temperature.mat** from TED and load it into MATLAB. The file includes a data structure name **temperature**. Explore the field names, size and content of the data structure. The field names include: **Year**, **Month**, **Day** which indicate the date when the temperature is recorded; **Loc** field is a data structure which has 2 fields of **SanDiego** and **RapidCity**.

You are asked to write the following 3 functions. The first function is to get the daily temperature for a given date and location. It should have the following header: **function [T] = get_daily_T(temperature, year, month, day, location)** where **temperature** is the data structure; **year**, **month**, **day** are number inputs indicating the date. **Location** is a string being either **'SanDiego'** or **'RapidCity'**.

The second function is to compute the monthly-averaged temperature for a given year, month and location. It should have the following header: **function [T] = get_monthly_T (temperature, year, month, location)**. The inputs are similar to those in the first function except the input **day** is absent. The output **T** is the temperature averaged over the input month of the input year at the input location.

The third function is to compute the yearly-averaged temperature for a given year and location. It should have the following header: **function [T] = get_yearly_T(temperature, year, location)**. The inputs are similar to those in the first function except the inputs **day** and **month** are absent. The output **T** is the temperature averaged over the input year at the input location.

All functions should have a description. In cases when the input date or location is not available, set the output temperature **T='The requested date or location is not available'** and use function **return** to exit the function.

(a) Get the temperature for Feb. 31^{st} 1999 in SanDiego, and put the answer in **p3a**.

(b) Get the temperature for Feb. 28^{th} 1999 in SanDiego, and put the answer in **p3b**.

(c) Get the temperature for Feb. 28^{th} 1999 in RapidCity, and put the answer in **p3c**.

(d) Get the monthly-averaged temperature for the month of Feb. in 1990 in RapidCity, and put the answer in **p3d**.

(e) Get the monthly-averaged temperature for the month of Feb. in 1995 in RapidCity, and put the answer in **p3e**.

(f-i) Create a data structure named **monthly_temperature** with 4 fields: **Year**, **Month**, **SanDiego** and **RapidCity** to store the monthly-averaged temperature for each month from 1995 to 2010 for the two locations. Set the following:

p3f=[monthly_temperature.Year];
p3g=[monthly_temperature.Month];
p3h=[monthly_temperature.SanDiego];
p3i=[monthly_temperature.RapidCity];

(j) Get the yearly-averaged temperature for the year of 1990 for RapidCity and put the

answer in **p3j**.

(k) Get the yearly-averaged temperature for the year of 1995 for RapidCity and put the answer in **p3k**.

(l-n) Create a data structure named **yearly_temperature** with 3 fields: **Year**, **SanDiego** and **RapidCity** to store the yearly-averaged temperature from 1995 to 2010 for the two locations. Set the following:

p3l=[yearly_temperature.Year];
p3m=[yearly_temperature.SanDiego];
p3n=[yearly_temperature.RapidCity];