W1005 - Fall 2014
Homework 5

- Due by Friday 4pm (Nov. 21st).
- See submission instructions.
- Always include your name and UNI at the top of your submitted files.

1. The following questions should be answered in the text file **hw5.txt** (word/pdf file is OK too).
   Generate a vector of random numbers by executing two commands:
   ```
   >> rng(10)
   >> V = floor(rand(1,8) *51)
   ```

   a) Show the steps that the *Insertion Sort* algorithm takes to sort V in *ascending* order (see slide #16 for an example).
   b) Show the steps that the *Selection Sort* algorithm takes to sort V in *descending* order (as in part a, show V after each iteration). Note: you cannot use the pseudocode verbatim (its written for ascending order).

2. Write a function '**isort**' which implements the Insertion Sort algorithm (ascending order) for numbers.

   a) Use the template posted on the webpage as your starting point. You can add code to the file in any line as necessary but do not modify the template.
   b) Use tic/toc functions to calculate the running time of your entire algorithm for input of size {1000,10000,100000}, and record the time in seconds in the file hw5.txt.

3. Write a function '**obj_cmp**' which compares two "objects" based on specified comparison criteria and returns the "better" object.

   a) The function has three input arguments (O1,O2,cmp) and returns a scalar value C (1 if O1 is better, -1 if O2 is better, 1 by default or if equal)
   b) O1/O2 are variables of some unknown type. 'cmp' is a function handle which we will apply to the objects. Do error checking (use

error function) to verify that 'cmp' is in fact a function handle (hint: 'help isa'), and that O1/O2 are variables of the same type (hint: 'help class'). You can assume there are no other errors.

c) Apply the handle (cmp) to each object and use the '>=' operator to determine which object is better (the larger the better). Set C appropriately.

4. Team Results in European Cup Matches:

a) Download the file **uefa.txt** from the course webpage which contains the results of over 500 European cup futbol matches in recent years. The file is in tabular form, where each line lists data for a particular match (or pair of matches), and the fields are separated by a single tab.

The fields are (in this order): team 1 name, team 1 country (abbr), team 2 name, team 2 country (abbr), game 1 result, game 2 result. Your task is to write a tool that allows users to search for particular teams/countries/matches. Below we implement the first part of the task (2nd part in HW6).

b) Write a program **get_match.m** that given a filename such as uefa.txt (input argument), opens the file in the format described, reads in the data, and saves a *structure array* that contains the matches.

Each match should be represented as a struct with field-value pairs as follows {team1 => "team 1 name", flag1 => "team 1 country", team2 => "team 2 name", flag2 => "team 2 country", game1 => "result of game 1", game2 => "result of game 2"}.

c) Keep the following notes in mind when writing the code:

Note 1: Use ***textscan*** to read the data from the file into a cell array.
Note 2: abbreviated country names are three letters long ("Eng" for England, "Fin" for Finland).
Note 3: it is always a good idea to lower-case all your strings (team/country names). Advanced Hint: you can use 'cellfun' with function handles to do this nicely, but a loop works too.

Note 4: some of the matches have only one result, in that case game 2 result would be denoted NA (not available).
Note 5: save your structure array (named ‘matches’) to a file called ‘**uefa.mat**’.

When you have finished running your function, you should obtain a structure array similar to this:

```
>> matches

ans =
695x1 struct array with fields:
   team1
   flag1
   team2
   flag2
   game1
   game2

>> matches(1)

ans =

    team1: 'fc-santa-coloma'
    flag1: 'and'
    team2: 'f91-dudelange’
    flag2: 'lux'
   game1: '0-2'
   game2: '0-2'
```

Your zip folder should include the following files:

```
        hw5.txt
        isort.m
        obj_cmp.m
        get_match.m
```