1 Cell Arrays

Cell arrays are like numeric arrays, but their elements can be any MATLAB construct, even another cell array. They are useful for combining disparate types of information into a single variable. For example, suppose you wanted to keep track of weather information, including the date, low and high temperatures, cloud cover (clear, partly cloudy, cloudy), and amount of precipitation. Here is an example of one way you could do that.

```matlab
weather_info = cell(3,4); %This creates an empty 3 by 4 cell array
weather_info(1,:) = {'July 1, 2015', [65, 78], 'Clear', 0};
weather_info(2,:) = {'July 2, 2015', [60, 72], 'Cloudy', 1};
weather_info(3,:) = {'July 3, 2015', [65, 83], 'Partly Cloudy', 0};
```

Try typing the following commands to see the difference in the output.

```matlab
weather_info(1,:)
weather_info{1,:}
```

2 Structures

Structures are similar to cell arrays, but elements are indexed by name rather than by numerical index. If you wanted to record the same information in a structure array that you recorded in the cell array `weather_info` here is one way you could do it.

```matlab
weather_data(3) = struct('date', 'July 3, 2015','temps',[65, 83], 'cover','partly cloudy', 'precip', 0);
% Starting with the last array entry causes MATLAB to preallocate space for the whole array
weather_data(2) = struct('date', 'July 2, 2015','temps',[60, 72], 'cover','cloudy', 'precip', 1);
weather_data(1) = struct('date', 'July 1, 2015','temps',[65, 78], 'cover','clear', 'precip', 0);
```

A second way to generate the structure `weather_data` given that the cell array `weather_info` already exists is to use the `cell2struct` command.
weather_data = cell2struct(weather_info, {'date','temps','cover','precip'}, 2);

Here is how you can reference the temps field in the structure weather_data(2):
weather_data(2).temps

You can also convert structures to cell arrays using the command struct2cell. For example, the following command would convert the structure weather_data to the cell array UML_weather_data:
UML_weather_data = struct2cell(weather_data);
Practice Problems

1. Create the following cell array \( ca = \{ \text{'abc'}, 11, 3:2:9, \text{zeros}(2,2) \} \)

   Use the \texttt{reshape} function to make \( ca \) a \( 2 \times 2 \) matrix. Then, write an expression that refers to the last column of this cell array (please reference Handout on Working with arrays).

2. Write a script file defining the following three cell array variables:
   \begin{verbatim}
   names = \{ 'Harry', 'Mike', 'Sue' \}
   verbs = \{ 'loves', 'eats' \}
   noun = \{ 'baseballs', 'rocks', 'sushi' \}
   \end{verbatim}

   Have your script print a sentence using one random element from each cell array (e.g. Mike eats sushi) (Hint: The command \texttt{rule = randi([1 3])} will generate a 1, 2, or 3 with equal probability and the command \texttt{rule = randi([1 2])} will generate a 1 or 2 with equal probability.)

3. Create an array of structures to store information on students’ quiz scores. Each structure in the array contains 4 numbers: the student’s ID number, quiz 1 score, quiz 2 score, and quiz 3 score. Use the following data:

<table>
<thead>
<tr>
<th>idNo</th>
<th>q1</th>
<th>q2</th>
<th>q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>7</td>
<td>7.5</td>
<td>8</td>
</tr>
<tr>
<td>33</td>
<td>5.5</td>
<td>6</td>
<td>6.5</td>
</tr>
<tr>
<td>37</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>24</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>