Question 4: (25 points)

Function `MakeRoom` is given for creating a *Room* struct (structure):

```
function R = MakeRoom(roomID,s)
% R is a struct such that
% R.id is assigned the string roomID, the name of the room, and
% R.seats is assigned the type double scalar s, the size (number of seats) of the room.
R = struct('id',roomID,'seats',s);
```

Implement the following function as specified by making effective use of function `MakeRoom` and built-in functions `strfind` and `str2double`. See the bottom of this page for an *example* cell array containing room data.

```
function Q = roomData(D)
% D is a 1-d cell array of strings; D is not empty.
% In each cell of D is a string:
%   - the string gives the data (id and size) of at least one room
%   - the string begins and ends with exactly one space
%   - exactly one space separates the room id and room size substrings
% The room id substring does not contain any spaces and the data for one room is
% always in the same cell (not split across two cells).
% Q is a 1-d struct array of Room structs that stores all the room data given in D.
```

Example solution:

```
r=0; % number of rooms processed so far

for i= 1:length(D)

    pos= strfind(D{i}, ' ');

    for k= 1:2:length(pos)-1  % -2 also ok when increment is 2
                              % Really just need to exclude last space

           id= D{i}(pos(k)+1:pos(k+1)-1);

           seats= str2double(D{i}(pos(k+1):pos(k+2)));

           r= r+1;
           Q(r)= MakeRoom(id, seats);
    end
end
```

For example, if the cell array of room data is

```
D= {' Gates314 47 KimballB11 190 '; ...
    ' GatesG100 155 Gates114 57 Gates122 55 '; ...
    ' MarthaVanRensselaerG150 85 '}
```

then `roomData(D)` should return an array of length 6, where each component is a Room struct and stores the data of one room. Recall that `str2double` can handle leading and trailing spaces.

Question 5: (20 points)

Complete the function below as specified.

```
function showRoomCombos(st, Q)
% Show all possible, unique, combinations of three rooms that can be used
%   by st students in a course for an exam.
% st is a positive integer.
% Q is an array of Room structs as specified in Problem 4 (see previous page).
% Q is not empty and each component of Q is one Room struct.
% Q contains all the rooms that are available for the exam.  Note that st
%   students require at least 2*st seats in total so that the students can
%   spread out.
% For each unique 3-room combination that can accommodate the exam, print one
%   line that shows the three room ids and the total number of seats.
% If no 3-room combination can accommodate the students, print 'Not enough seats'.
% Be efficient for full credit--avoid unnecessary iterations.
```

Example solution:

```
enuf= 0;

n= length(Q);

for i= 1:n-2  % going to n is ok

    for j= i+1:n-1  % going to n is ok

        for k= j+1: n

            total= Q(i).seats + Q(j).seats + Q(k).seats;

            if total >= 2*st

                fprintf('%s %s %s %d\n', Q(i).id, Q(j).id, Q(k).id, total)

                enuf= enuf + 1;  % not necessary to count; set to nonzero is ok
            end
        end
    end
end

if enuf==0

    disp('Not enough seats')
end
```