```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function receives a table of grades for different assignments with %
% the student IDs and their names and returns the same table, arranged    %
% alphabetically, with an extra column with the calculated final grades,  %
% called "Final"                                                          %
%                                                                         %
% Input:  A table containing grades on the 7-step-scale given to          %
%         N students on M different assignments.                          %
% Output: The same table with an added extra column containing the        %
%         final grades                                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function finalTable = displayListOfGrades(Table)

% Extract the grades from the table based on row number, assuming that each
% table starts with a StudentID and a Name
grades = Table{: , 3:width(Table)};

% Use the computeFinalGrades.m function to compute the final grades
Final = computeFinalGrades(grades);

% Reshape Final into a matrix
Final = reshape(Final , length(Final) , 1);

% Turn the vector 'Final' into a table
averageGrades = table(Final);

% Combine 'Final' with the initial table
tableWithAverage = [Table , averageGrades];

% Set a new variable equal to 'tableWithAverage'
tableLow = tableWithAverage;

% make all letters lower case in the "Name" column
tableLow.Name = lower(tableLow.Name);

% Create variable 'b' equal to the table with sorted rows and variable 'I'
% equal to the references of the old table with relation to the sorted
% table
[~ , I] = sortrows(tableLow , 'Name');

% The result is equal to 'tableWithAverage', rearranged according to the
% references in 'I'
finalTable = tableWithAverage(I , :);
```