Table of Contents

clear

```
clear all;
clc;
```

Problem 1:

Part 1:

```
load HW6.mat

% Part 2:
avgRacingInfo = mean(RacingInfo, 2); % Calc avg column
concatenatedRacingInfo = [RacingInfo avgRacingInfo]; % concatenate to
 last column
fprintf('Problem 1 Part 2:\n');
disp(concatenatedRacingInfo); % Display the concatenated matrix.

% Part 3:
sortedAvg = sortrows(concatenatedRacingInfo,
 size(concatenatedRacingInfo, 2)); % Sort based on avg
fprintf('Problem 1 Part 3:\n');
disp(sortedAvg); % Display the sorted matrix.
a = sortedAvg(1, 1); % The fastest one
b = sortedAvg(2, 1); % The second fastest one
fprintf('The number of the two dogs will be selected to the race are
 %d and %d.\n', a, b); % Use fprintf() to display which dogs will be
 selected to the race.
```

```
Problem 1 Part 2:
    1.0000   10.0000   13.0000   14.0000   12.0000   19.0000   11.5000
    2.0000    5.0000   19.0000   20.0000   23.0000   10.0000   13.1667
    3.0000   22.0000    7.0000    9.0000    7.0000    8.0000    9.3333
    4.0000   13.0000   18.0000   11.0000    9.0000    9.0000   10.6667
    5.0000   10.0000   11.0000   10.0000    9.0000    8.0000    8.8333

Problem 1 Part 3:
    5.0000   10.0000   11.0000   10.0000    9.0000    8.0000    8.8333
    3.0000   22.0000    7.0000    9.0000    7.0000    8.0000    9.3333
    4.0000   13.0000   18.0000   11.0000    9.0000    9.0000   10.6667
    1.0000   10.0000   13.0000   14.0000   12.0000   19.0000   11.5000
    2.0000    5.0000   19.0000   20.0000   23.0000   10.0000   13.1667
```

The number of the two dogs will be selected to the race are 5 and 3.

Problem 2:

```
%Part 1:
N = 5;
matrixPyramid = Pyramid(N);
fprintf('Problem 2 Part 1:\n');
disp(matrixPyramid); % Run and display sample

%Part 2:
N = 5;
matrixVshape = Vshape(N);
fprintf('Problem 2 Part 2:\n');
disp(matrixVshape); % Run and display sample

%Part 3:
N = 5;
matrixReflect = Reflect(N);
fprintf('Problem 2 Part 3:\n');
disp(matrixReflect); % Run and display sample
```

```
Problem 2 Part 1:
     0     0     0     0     1     0     0     0     0
     0     0     0     2     1     2     0     0     0
     0     0     3     2     1     2     3     0     0
     0     4     3     2     1     2     3     4     0
     5     4     3     2     1     2     3     4     5

Problem 2 Part 2:
     1     0     0     0     0     0     0     0     1
     1     2     0     0     0     0     0     2     1
     1     2     3     0     0     0     3     2     1
     1     2     3     4     0     4     3     2     1
     1     2     3     4     5     4     3     2     1

Problem 2 Part 3:
     6     4     3     2     1
     1     6     3     2     1
     1     2     6     2     1
     1     2     3     6     1
     1     2     3     4     6
```

Problem 3:

Part 1: a):

```
avgHw = mean(hwScore(:,2:size(hwScore, 2)), 2); % Calc avg hw score of
 each student
updatedHwScore = hwScore; % Set a new updated matrix
for i = 1:size(hwScore, 1)
    for j = 2:size(hwScore, 2)
```

```
        if (hwScore(i, j) < 0 | hwScore(i, j) > 10)
            updatedHwScore(i, j) = avgHw(i, 1);
        end
    end
end
% Walk through the whole matrix find illeagel cells and replace them
 with
% correspondent average value

avgMid = mean(midScore(:,2:size(midScore, 2)), 2); % Calc avg mid
 score of each student
updatedMidScore = midScore; % Set a new updated matrix
for i = 1:size(midScore, 1)
    for j = 2:size(midScore, 2)
        if (midScore(i, j) < 0 | midScore(i, j) > 30)
            updatedMidScore(i, j) = avgMid(i, 1);
        end
    end
end
% Walk through the whole matrix find illeagel cells and replace them
 with
% correspondent average value

avgFinal = mean(finalScore(:,2:size(finalScore, 2)), 2); % Calc avg
 final score of each student
updatedFinalScore = finalScore; % Set a new updated matrix
for i = 1:size(finalScore, 1)
    for j = 2:size(finalScore, 2)
        if (finalScore(i, j) < 0 | finalScore(i, j) > 50)
            updatedFinalScore(i, j) = avgFinal(i, 1);
        end
    end
end
% Walk through the whole matrix find illeagel cells and replace them
 with
% correspondent average value

fprintf('Problem 3 Part 1.a:\n'); % Print with format
disp(updatedHwScore);
disp(updatedMidScore);
disp(updatedFinalScore);
% Display the updated matrices.

% b):
allScores = [updatedHwScore updatedMidScore(:, 2:
 size(updatedMidScore, 2)) updatedFinalScore(:, 2:
 size(updatedFinalScore, 2))]; % make allScores matrix
fprintf('Problem 3 Part 1.b:\n'); % Print with format
disp(allScores); % Display allScores.

% c):
fullStatus = (allScores(:, 2: size(allScores, 2)) == 0); % determine
 which nodes are zeros
rowStatus = all(fullStatus, 2); % mart the rows full of 0 as 1(true)
```

```
newAllScore = allScores; % Set a new all Score matrix
for i = 1:size(newAllScore, 1)
    if(rowStatus(i, 1) == 1)
        newAllScore(i, :) = []; % delete row
    end
end
fprintf('Problem 3 Part 1.c:\n'); % Print with format
disp(newAllScore); % Display new allScores.

% Part 2:
% a)
roundAvgHw = avgHw; % use the avg hw score calced before
for i = 1:size(roundAvgHw, 1)
    roundAvgHw(i, 1) = round(roundAvgHw(i, 1)); % round
end
roundAvgHw = [hwScore(:, 1) roundAvgHw];

roundAvgMid = avgMid; % use the avg midterm score calced before
for i = 1:size(roundAvgMid, 1)
    roundAvgMid(i, 1) = round(roundAvgMid(i, 1)); % round
end
roundAvgMid = [midScore(:, 1) roundAvgMid];

fprintf('Problem 3 Part 2.a:\n'); % Print with format
disp(roundAvgHw);
disp(roundAvgMid);
% Display the results.

% b):
sumAdjustedAllScore = sum(allScores(:, 2:size(allScores, 2)), 2);
sumOutOf100(:, 1) = (sumAdjustedAllScore(:, 1) + 10) / (10 + 10
 * (size(hwScore, 2) - 1) + 50 * (size(finalScore, 2) - 1) + 10 *
 (size(hwScore, 2) - 1)) * 100; % Calc score out of 100
grades = [allScores sumOutOf100]; % Generate grades
for i = size(grades, 1): -1 : 2
    if(grades(i, size(grades, 2)) == 6.25)
        grades(i, :) = []; % delete row
    end
end
% readjust
fprintf('Problem 3 Part 2.b:\n'); % Print with format
disp(grades); % Display the results.

% c):
G = [grades(:, 1)]; % generate empty G with only student number
for i = 1:size(grades, 1)
    if grades(i, size(grades, 2)) >= 90
        G(i, 2) = num2str('A');
    elseif grades(i, size(grades, 2)) >= 80
        G(i, 2) = num2str('B');
    elseif grades(i, size(grades, 2)) >= 70
        G(i, 2) = num2str('C');
    elseif grades(i, size(grades, 2)) >= 60
        G(i, 2) = num2str('D');
```

```
    else
        G(i, 2) = num2str('E');
    end
end
fprintf('Problem 3 Part 2.c:\n'); % Print with format
disp(G); % Display the results.
```

```
Problem 3 Part 1.a:
  101.0000    5.0000    6.0000    6.0000    8.0000    7.0000
  102.0000    7.0000    8.0000    6.0000    9.0000    7.0000
  103.0000    7.0000    5.0000    6.0000    8.0000    8.0000
  104.0000    5.0000    6.0000    6.0000    7.0000    6.0000
  105.0000         0         0         0         0         0
  106.0000    4.0000    7.0000    5.0000    4.0000    6.0000
  107.0000    6.0000    7.0000    9.0000    6.0000    8.0000
  108.0000    6.0000    7.0000    9.0000    6.0000    8.0000
  109.0000    6.0000    7.0000    5.0000         0    8.0000
  110.0000    5.0000    7.0000    6.0000    8.0000    6.0000
  111.0000    9.0000    8.0000    6.0000    5.0000    4.2000

  101.0000   15.0000   26.0000   18.0000
  102.0000   23.0000   25.0000   26.6667
  103.0000   17.0000   25.0000   26.0000
  104.0000   25.0000   16.0000   26.0000
  105.0000         0         0         0
  106.0000   17.0000   24.0000   26.0000
  107.0000   28.0000   26.0000   22.0000
  108.0000   19.0000   26.0000   28.0000
  109.0000   27.0000   21.0000   26.0000
  110.0000   21.0000   20.0000   26.0000
  111.0000   21.0000   20.0000   26.0000

   101    47
   102    28
   103    33
   104    35
   105     0
   106    26
   107    42
   108    38
   109    41
   110    39
   111    31

Problem 3 Part 1.b:
  Columns 1 through 7

  101.0000    5.0000    6.0000    6.0000    8.0000    7.0000   15.0000
  102.0000    7.0000    8.0000    6.0000    9.0000    7.0000   23.0000
  103.0000    7.0000    5.0000    6.0000    8.0000    8.0000   17.0000
  104.0000    5.0000    6.0000    6.0000    7.0000    6.0000   25.0000
  105.0000         0         0         0         0         0         0
  106.0000    4.0000    7.0000    5.0000    4.0000    6.0000   17.0000
  107.0000    6.0000    7.0000    9.0000    6.0000    8.0000   28.0000
```

```
  108.0000    6.0000    7.0000    9.0000    6.0000    8.0000   19.0000
  109.0000    6.0000    7.0000    5.0000         0    8.0000   27.0000
  110.0000    5.0000    7.0000    6.0000    8.0000    6.0000   21.0000
  111.0000    9.0000    8.0000    6.0000    5.0000    4.2000   21.0000

  Columns 8 through 10

   26.0000   18.0000   47.0000
   25.0000   26.6667   28.0000
   25.0000   26.0000   33.0000
   16.0000   26.0000   35.0000
         0         0         0
   24.0000   26.0000   26.0000
   26.0000   22.0000   42.0000
   26.0000   28.0000   38.0000
   21.0000   26.0000   41.0000
   20.0000   26.0000   39.0000
   20.0000   26.0000   31.0000

Problem 3 Part 1.c:
  Columns 1 through 7

  101.0000    5.0000    6.0000    6.0000    8.0000    7.0000   15.0000
  102.0000    7.0000    8.0000    6.0000    9.0000    7.0000   23.0000
  103.0000    7.0000    5.0000    6.0000    8.0000    8.0000   17.0000
  104.0000    5.0000    6.0000    6.0000    7.0000    6.0000   25.0000
  106.0000    4.0000    7.0000    5.0000    4.0000    6.0000   17.0000
  107.0000    6.0000    7.0000    9.0000    6.0000    8.0000   28.0000
  108.0000    6.0000    7.0000    9.0000    6.0000    8.0000   19.0000
  109.0000    6.0000    7.0000    5.0000         0    8.0000   27.0000
  110.0000    5.0000    7.0000    6.0000    8.0000    6.0000   21.0000
  111.0000    9.0000    8.0000    6.0000    5.0000    4.2000   21.0000

  Columns 8 through 10

   26.0000   18.0000   47.0000
   25.0000   26.6667   28.0000
   25.0000   26.0000   33.0000
   16.0000   26.0000   35.0000
   24.0000   26.0000   26.0000
   26.0000   22.0000   42.0000
   26.0000   28.0000   38.0000
   21.0000   26.0000   41.0000
   20.0000   26.0000   39.0000
   20.0000   26.0000   31.0000

Problem 3 Part 2.a:
   101     6
   102     7
   103     7
   104     6
   105     0
   106     5
   107     7
```

```
   108     7
   109     5
   110     6
   111     4

   101    20
   102    27
   103    23
   104    22
   105     0
   106    22
   107    25
   108    24
   109    25
   110    22
   111    22

Problem 3 Part 2.b:
  Columns 1 through 7

  101.0000    5.0000    6.0000    6.0000    8.0000    7.0000   15.0000
  102.0000    7.0000    8.0000    6.0000    9.0000    7.0000   23.0000
  103.0000    7.0000    5.0000    6.0000    8.0000    8.0000   17.0000
  104.0000    5.0000    6.0000    6.0000    7.0000    6.0000   25.0000
  106.0000    4.0000    7.0000    5.0000    4.0000    6.0000   17.0000
  107.0000    6.0000    7.0000    9.0000    6.0000    8.0000   28.0000
  108.0000    6.0000    7.0000    9.0000    6.0000    8.0000   19.0000
  109.0000    6.0000    7.0000    5.0000         0    8.0000   27.0000
  110.0000    5.0000    7.0000    6.0000    8.0000    6.0000   21.0000
  111.0000    9.0000    8.0000    6.0000    5.0000    4.2000   21.0000

  Columns 8 through 11

   26.0000   18.0000   47.0000   92.5000
   25.0000   26.6667   28.0000   93.5417
   25.0000   26.0000   33.0000   90.6250
   16.0000   26.0000   35.0000   88.7500
   24.0000   26.0000   26.0000   80.6250
   26.0000   22.0000   42.0000  102.5000
   26.0000   28.0000   38.0000   98.1250
   21.0000   26.0000   41.0000   94.3750
   20.0000   26.0000   39.0000   92.5000
   20.0000   26.0000   31.0000   87.6250

Problem 3 Part 2.c:
   101    65
   102    65
   103    65
   104    66
   106    66
   107    65
   108    65
   109    65
   110    65
```

```
111     66
```

Published with MATLAB® R2015b