

Assignment 6 Solution (CEE 3804)

Problem 1:

Task 1)

```
Editor - G:\My Drive\Semester 2\Computer Application in CEE- TA\Assignment 6\BusCompany.m
BusCompany.m × + Task 1/ Rroblem 1/Assignment 6/CEE 3804
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %% Import data from spreadsheet
3 % Script for importing data from the following spreadsheet:
4 %
5 %   Workbook: G:\My Drive\Semester 2\Computer Application in CEE- TA\Assignment 6\BusCompany_Data.xlsx
6 %   Worksheet: Bus Data
7 %
8 % Auto-generated by MATLAB on 21-Mar-2021 22:56:28
9
10 -clic
11 -clear
12 -close all
13
14 %% Setup the Import Options and import the data
15 opts = spreadsheetImportOptions("NumVariables", 5);
16
17 % Specify sheet and range
18 opts.Sheet = "Bus Data";
19 opts.DataRange = "A2:E742";
20
21 % Specify column names and types
22 opts.VariableNames = ["City", "BusType", "Age", "Miles", "Routelength"];
23 opts.VariableTypes = ["categorical", "categorical", "double", "double", "double"];
24
25 % Specify variable properties
26 opts = setvarsopts(opts, ["City", "BusType"], "EmptyFieldRule", "auto");
27
28 % Import the data
29 BusCompanyData = readable("G:\My Drive\Semester 2\Computer Application in CEE- TA\Assignment 6\BusCompany_Data.xlsx", opts,
30
31
32 %% Clear temporary variables
33 clear opts
34
35 %%Difene the headers of the inpute table as variable to work on.
36 city= BusCompanyData.City;
37 bustype= BusCompanyData.BusType;
38 age= BusCompanyData.Age;
39 milage= BusCompanyData.Miles;
40 routelength= BusCompanyData.Routelength;
41
42
```

```
Editor - G:\My Drive\Semester 2\Computer Application in CEE- TA\Assignment 6\BusCompany.m
BusCompany.m × + Editor - G:\My Drive\Semester 2\Computer Application in CEE- TA\Assignment 6\BusCompany.m
BusCompany.m × + Task 1/ Rroblem 1/Assignment 6/CEE 3804
31
32 %% Clear temporary variables
33 clear opts
34
35 %%Difene the headers of the inpute table as variable to work on.
36 city= BusCompanyData.City;
37 bustype= BusCompanyData.BusType;
38 age= BusCompanyData.Age;
39 milage= BusCompanyData.Miles;
40 routelength= BusCompanyData.Routelength;
41
42
43 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Task 2/ Problem 1%%%%%%%%%%%%%
44 bus_city_seattle= find (city== 'Seattle'); %find 'seattle' among the cities in the inpute table.
45 number_of_bus_seattle= length (bus_city_seattle); %Count number of buses in seattle.
46
47 routelength_seattle= routelength(bus_city_seattle); % find the route length of corresponding city.
48 average_milage_seattle= mean(routelength_seattle); %calculate the average route length in seattle.
49
50
51 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Task 3/ Problem 1%%%%%%%%%%%%%
52 bus_city_Atlanta= find (city== 'Atlanta'); %find 'Atlanta' among the cities in the inpute table.
53 routelength_Atlanta= routelength(bus_city_Atlanta); % find the route length of corresponding city.
54 histogram(routelength_Atlanta)% draw histogram diagram
55 title('Route Length Buses in Atlanta')
56 xlabel('Route Length (miles)')
57 ylabel('Number of Buses')
58
59
60 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Task 4/ Problem 1%%%%%%%%%%%%%
61 bus_newflyer = find (bustype == 'New Flyer XDE40'); %find 'New Flyer XDE40' among the bus types in the inpute table.
62 age_newflyer= age(bus_newflyer); %find the age of corresponding bus type.
63 average_age_newflyer = mean(age_newflyer); %calculate the average age of 'New Flyer XDE40'.
64
65
66 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Task 5/ Problem 1%%%%%%%%%%%%%
67 bus_city_SaltLakeCity = find (city == 'Salt Lake City'); %find 'Salt Lake City' among the cities in the inpute table.
68 age_SaltLakeCity = age(bus_city_SaltLakeCity); %find the age of corresponding city.
69 milage_SaltLakeCity = milage(bus_city_SaltLakeCity); % find the milage of corresponding city.
70 dlmwrite('Bus Selected Characteristics in Salt Lake City.txt',[age_SaltLakeCity,milage_SaltLakeCity], 'precision', '%.0f', '
```

Task 2 and 4)

Application in CEE- TA ▶ Assignment 6

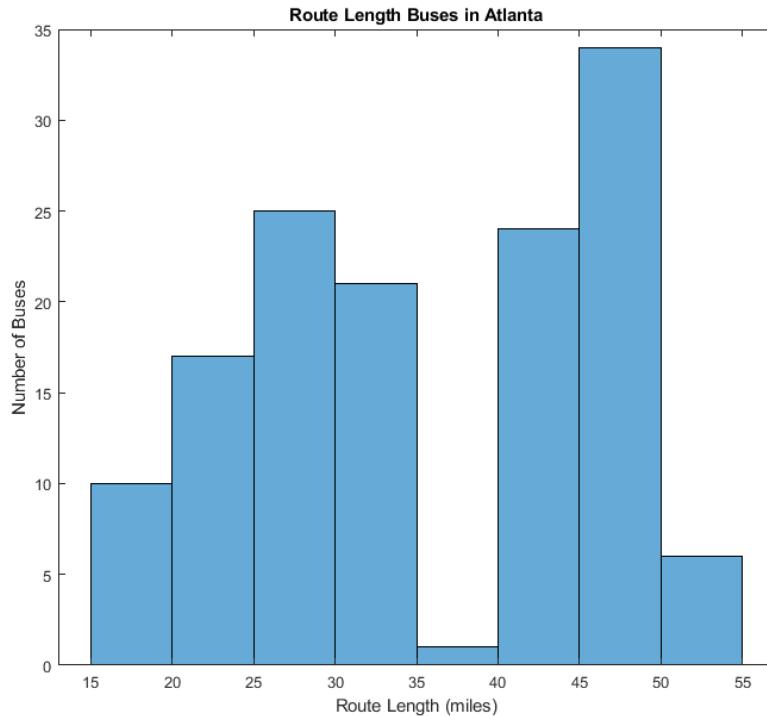
The screenshot shows the MATLAB environment. The Editor window displays a script named 'BusCompany.m' with the following code:

```
41
42 %%%%%%% Task 2 / Problem 1%%%%%%
43 bus_city_seattle= find (city== 'Seattle'); %find 'seattle' among the cities in the inpute table.
44 number_of_bus_seattle= length (bus_city_seattle); %Count number of buses in seattle.
45
46 routelength_seattle= routelength(bus_city_seattle); % find the route length of corresponding city.
47 average_milage_seattle= mean(routelength_seattle); %calculate the average route length in seattle.
```

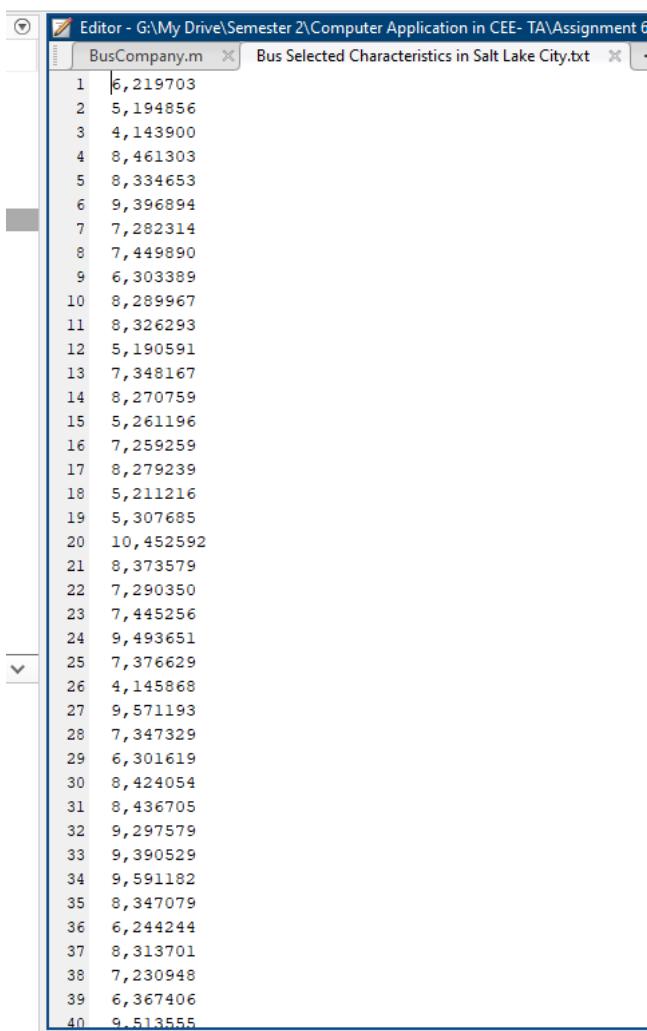
The Command Window shows the results of running the script:

```
>> number_of_bus_seattle
number_of_bus_seattle =
189
>> average_milage_seattle
average_milage_seattle =
35.8210
>> average_age_newflyer
average_age_newflyer =
7.2045
fx >>
```

Task 3)



Task 5)



The screenshot shows a code editor window with two tabs. The active tab, 'BusCompany.m', contains the following code:

```
1 6,219703
2 5,194856
3 4,143900
4 8,461303
5 8,334653
6 9,396894
7 7,282314
8 7,449890
9 6,303389
10 8,289967
11 8,326293
12 5,190591
13 7,348167
14 8,270759
15 5,261196
16 7,259259
17 8,279239
18 5,211216
19 5,307685
20 10,452592
21 8,373579
22 7,290350
23 7,445256
24 9,493651
25 7,376629
26 4,145868
27 9,571193
28 7,347329
29 6,301619
30 8,424054
31 8,436705
32 9,297579
33 9,390529
34 9,591182
35 8,347079
36 6,244244
37 8,313701
38 7,230948
39 6,367406
40 9,513555
```

Problem 2:

Task 1)

The screenshot shows the MATLAB Editor window with the file 'Train_Leg.m' open. The code calculates the equivalent noise level (Leq) generated by a train based on input parameters: SELref (reference sound exposure level), Ncars (number of cars in the train), S (train speed), and V (hourly average train volume). It also finds the index of Leq values less than 60 dB and the maximum Leq value corresponding to the highest speed.

```
Editor - G:\My Drive\Semester 2\Computer Application in CEE- TA\Assignment 6\Train_Leg.m
Train_Leg.m + Task 1 and 2/ problem 2/Assignment 6/CEE 3804%%%%%
1 %%%%%%Task 1 and 2/ problem 2/Assignment 6/CEE 3804%%%%%
2 % script calculates the noise generated by a train
3 -
4 - clc
5 - clear
6 - close all
7 -
8 - % formula to estimate the noise generated by a train is found to be:
9 - % Leg = SELref + 10 log(Ncars) + 20 log(S/50) + 10 log (V) - 31.6
10 - % where:
11 - % Leg = equivalent noise level (decibels - dBA)
12 - % SEL ref = reference sound exposure level (decibels - dBA)
13 - % Ncars = number of cars in the train
14 - % S = train speed (mph)
15 - % V = hourly average train volume (trains per hour)
16 - % log = natural log of the number
17 -
18 - %Define inpute parameters
19 - SELref = 55;
20 - Ncars = 10;
21 - S = (10:1:60);
22 - V = 25;
23 - %Calculate noise generated by the train
24 - Leg = SELref + 10 * log(Ncars) + 20 * log(S/50) + 10 * log(V) - 31.6;
25 -
26 - %%%%%%Task 3/ problem 2/Assignment 6/CEE 3804%%%%%
27 - %find Leg values that are less than 60 dBL
28 - leg_less_60_index = find(Leg < 60);
29 - leg_less_60 = Leg(leg_less_60_index);
30 -
31 - %%%%%%Task 4/ problem 2/Assignment 6/CEE 3804%%%%%
32 - %find the speed of train correspondng to maximum Leg that is less than 60
33 - %dBL
34 - max_Leg_less_60 = max(leg_less_60);
35 - max_Leg_less_60_index= find(Leg == max_Leg_less_60 );
36 - Max_S = S(max_Leg_less_60_index);
37 -
38 --
```

Task 2, 3, and 4)

The screenshot shows the MATLAB Command Window displaying the results of the script execution. It shows the generated Leq matrix, the index of values less than 60, the maximum Leq value, and the corresponding speed.

```
Command Window
>> Leg
Leg =
Columns 1 through 13
46.4259 48.3321 50.0723 51.6731 53.1553 54.5352 55.8259 57.0384 58.1816 59.2629 60.2888 61.2646 62.1950
Columns 14 through 26
63.0840 63.9352 64.7517 65.5361 66.2909 67.0182 67.7201 68.3981 69.0539 69.6889 70.3043 70.9014 71.4811
Columns 27 through 39
72.0445 72.5925 73.1259 73.6454 74.1517 74.6456 75.1275 75.5982 76.0579 76.5074 76.9470 77.3771 77.7982
Columns 40 through 51
78.2106 78.6146 79.0107 79.3990 79.7800 80.1538 80.5208 80.8812 81.2352 81.5830 81.9249 82.2610
>> leg_less_60
leg_less_60 =
46.4259 48.3321 50.0723 51.6731 53.1553 54.5352 55.8259 57.0384 58.1816 59.2629
>> Max_S
Max_S =
19
fx >>
```

Problem 3:

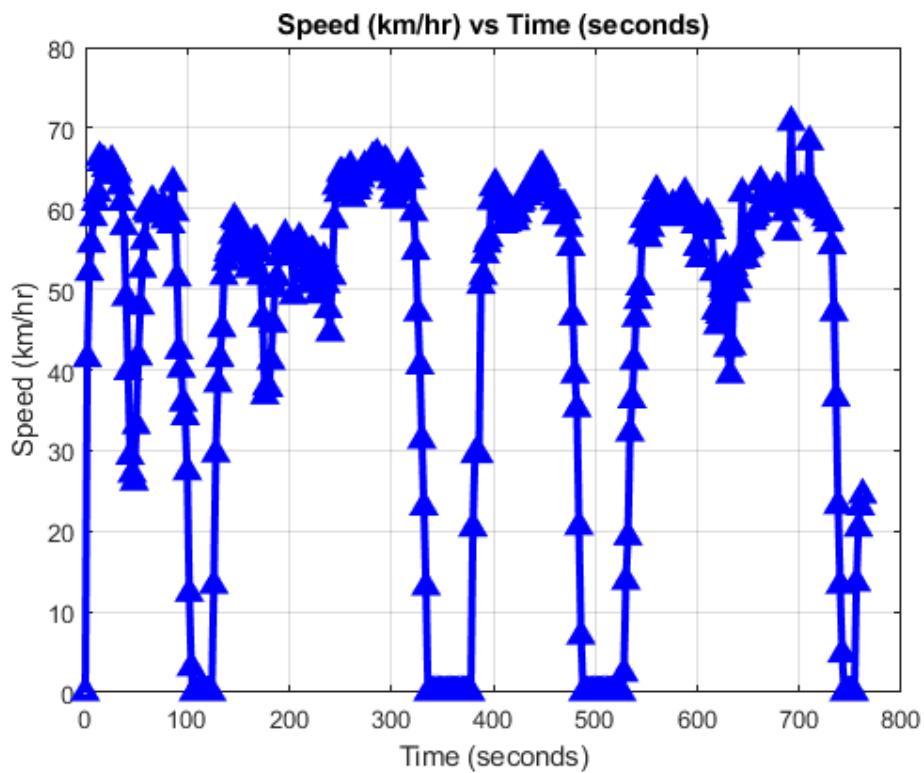
Task 1)

```

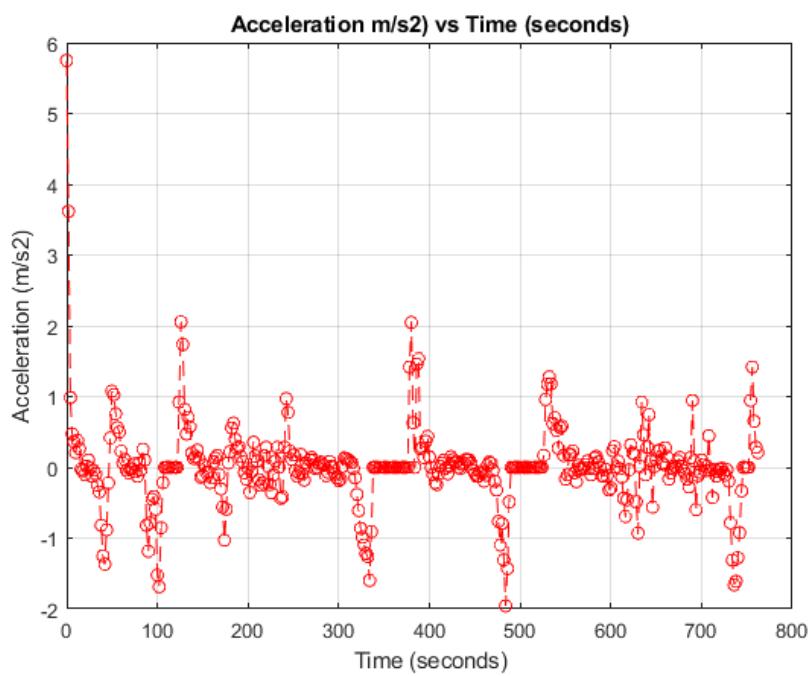
Car_GPS.m  % U T %
1 %%%%%%%%%%%%%%Task 1/ problem 3/Assignment 6/CEE 3804%%%%%%%%%%%%%
2 % script to read GPS car data collected in rural roads in Arizona
3 -
4 - clc
5 - clear
6 - close all
7 -
8 - % load the GPS data file to WorkSpace.
9 - load ('GPS_Data.m')
10 -
11 - % columns of the GPS data file
12 - % Column 1 = Time of observation (seconds)
13 - % Column 2 = Distance traveled (m)
14 - % Column 3 = Speed (km/hr)
15 - % Column 4 = Acceleration (m/s-s)
16 -
17 - %%%%%%%%%%%%%%Task 2/ problem 3/Assignment 6/CEE 3804%%%%%%%%%%%%%
18 - % define and obtain the parameter named speed and time in GPS data file.
19 - time = GPS_Data(:,1);
20 - speed = GPS_Data(:,3);
21 - % Plot the speed of the car (in y-axis) vs time (x- axis).
22 - figure
23 - plot(time,speed, '^-b','LineWidth',3)
24 - title('Speed (km/hr) vs Time (seconds)')
25 - xlabel('Time (seconds)')
26 - ylabel('Speed (km/hr)')
27 - grid
28 -
29 - hold on
30 - %%%%%%%%%%%%%%Task 3/ problem 3/Assignment 6/CEE 3804%%%%%%%%%%%%%
31 - % convert km/hr to m/s
32 - speed_meter_second = speed .* (1000/3600);
33 - % calculate the acceleration rate of car in m/s2
34 - acceleration = gradient(speed_meter_second,time);
35 -
36 - % Plot the acceleration of the car (in y-axis) vs time (x- axis).
37 - figure
38 - plot(time,acceleration, 'o--r')
39 - title('Acceleration m/s2) vs Time (seconds)')
40 - xlabel('Time (seconds)')
41 - ylabel('Acceleration (m/s2)')
42 - grid
43 -
44 - hold on
45 - %%%%%%%%%%%%%%Task 4/ problem 3/Assignment 6/CEE 3804%%%%%%%%%%%%%
46 - min_acceleration = min(acceleration); %find the maximum deceleration that is equal to minimum acceleration in acceleration v
47 - min_acceleration_index = find(acceleration == min_acceleration ); %find the index of minimum acceleration.
48 - min_acceleration_time = time(min_acceleration_index); %find the time at which, vehicle has the maximum deceleration.
49 -
50 -
51 - %%%%%%%%%%%%%%Task 5/ problem 3/Assignment 6/CEE 3804%%%%%%%%%%%%%
52 - %Find the average speed of the car for the complete profile.
53 - speed_average= mean(speed);
54 -
55 -
56 - %%%%%%%%%%%%%%Task 6/ problem 3/Assignment 6/CEE 3804%%%%%%%%%%%%%
57 - %find the number of seconds the car is stopped during the recorded data.
58 - %Time interval between successive observations in data is 2 seconds.
59 - stop_time_index = find(speed == 0);
60 - stop_time_duration= length(stop_time_index)*2;

```

Task 2)



Task 3)



Task 4, 5, and 6)

The screenshot shows the MATLAB environment. The Editor window at the top contains the script `Car_GPS.m` with the following code:

```
38 - plot(time,acceleration, 'o--r')
39 - title('Acceleration m/s2) vs Time (seconds)')
40 - xlabel('Time (seconds)')
41 - ylabel('Acceleration (m/s2)')
42 - grid
```

The Command Window below displays the results of running the script:

```
>> min_acceleration
min_acceleration =
-1.9583

>> min_acceleration_time
min_acceleration_time =
484

>> speed_average
speed_average =
45.5246

>> stop_time_duration
stop_time_duration =
118

fx >>
```

Problem 4:

Task 1)

The image shows two side-by-side MATLAB code snippets in the MATLAB Editor. Both snippets are titled 'Cruise_ships.m'. The left snippet is labeled 'Task 1)' and the right snippet is labeled 'Task 2)'. Both snippets are part of a larger script for 'Assignment 6/CEE 3804'.

Task 1) Code:

```
1 %%%%%% Task 1/ problem 4/Assignment 6/CEE 3804%%%%%
2 % script to read Cruise_ships data.
3 -
4 - clear
5 - close all
6 [num,txt,raw] = xlsread('Cruise_ships.xlsx');
7 -
8 %Define the name and variable for the parameters in the Cruise_ships data
9 number_of_ships= length(raw);
10 ship_name = raw(2:number_of_ships,1);
11 ship_tonnage = num( :,1);
12 ship_length = num( :,2);
13 ship_width = num( :,3);
14 ship_passengers = num( :,4);
15 ship_cabins = num( :,5);
16 -
17 %Find number of ships with tonnage greater than 200,000 tons.
18 ships_tonnage200000_number = length(find(ship_tonnage > 200000));
19 %Find the names of the ships with tonnage greater than 200,000 tons.
20 ships_tonnage200000_name = ship_name(find(ship_tonnage > 200000));
21 -
22 %%%%%% Task 2/ problem 4/Assignment 6/CEE 3804%%%%%
23 %find the number of the cruise ships with total length below 325 meters.
24 ships_length325_number = length(find(ship_length < 325));
25 %find the index of ship with length below 325 meters.
26 ships_length325_index= find(ship_length < 325);
27 %Find the names of the ships with total length below 325 meters.
28 ships_length325_name = ship_name(ships_length325_index);
29 -
30 %%%%%% Task 3/ problem 4/Assignment 6/CEE 3804%%%%%
31 %Find the average width of the ships in the database.
32 ship_width_average = mean(ship_width);
33 -
34 %%%%%% Task 4/ problem 4/Assignment 6/CEE 3804%%%%%
35 %find the cruise ships with a passenger to cabin ratio above 2.38.
36 ships_ratio= ship_passengers./ship_cabins;
37 ships_ratio_index = find(ships_ratio > 2.38);
38 ships_ratio_name = ship_name(ships_ratio_index);
39 -
40 %%%%%% Task 5/ problem 4/Assignment 6/CEE 3804%%%%%
41 %retrieve the tonnage and width of ships with a passenger to cabin ratio above 2.38.
42 ships_ratio_tonnage = ship_tonnage(ships_ratio_index);
43 ships_ratio_width = ship_width(ships_ratio_index);
44 -
45 %write name, tonnage, and width of the ships with a passenger to cabin ratio above 2.38 in a txt file using fprintf command.
46 fid = fopen ('ships with a passenger to cabin ratio above 2.38.txt','a');
47 fprintf(fid,'%s,%f,%f\n',ships_ratio_name(i),ships_ratio_tonnage(i),ships_ratio_width(i));
48 -
49 for i=1: length(ships_ratio_index)
50     fprintf(fid,'%s,%f,%f\n',ships_ratio_name(i),ships_ratio_tonnage(i),ships_ratio_width(i));
51 end
52 -
53 fclose(fid);
```

Task 2) Code:

```
1 %%%%%% Task 2/ problem 4/Assignment 6/CEE 3804%%%%%
2 %find the number of the cruise ships with total length below 325 meters.
3 ships_length325_number = length(find(ship_length < 325));
4 %find the index of ship with length below 325 meters.
5 ships_length325_index= find(ship_length < 325);
6 %Find the names of the ships with total length below 325 meters.
7 ships_length325_name = ship_name(ships_length325_index);
8 -
9 %%%%%% Task 3/ problem 4/Assignment 6/CEE 3804%%%%%
10 %Find the average width of the ships in the database.
11 ship_width_average = mean(ship_width);
12 -
13 %%%%%% Task 4/ problem 4/Assignment 6/CEE 3804%%%%%
14 %find the cruise ships with a passenger to cabin ratio above 2.38.
15 ships_ratio= ship_passengers./ship_cabins;
16 ships_ratio_index = find(ships_ratio > 2.38);
17 ships_ratio_name = ship_name(ships_ratio_index);
18 -
19 %%%%%% Task 5/ problem 4/Assignment 6/CEE 3804%%%%%
20 %retrieve the tonnage and width of ships with a passenger to cabin ratio above 2.38.
21 ships_ratio_tonnage = ship_tonnage(ships_ratio_index);
22 ships_ratio_width = ship_width(ships_ratio_index);
23 -
24 %write name, tonnage, and width of the ships with a passenger to cabin ratio above 2.38 in a txt file using fprintf command.
25 fid = fopen ('ships with a passenger to cabin ratio above 2.38.txt','a');
26 fprintf(fid,'%s,%f,%f\n',ships_ratio_name(i),ships_ratio_tonnage(i),ships_ratio_width(i));
27 -
28 for i=1: length(ships_ratio_index)
29     fprintf(fid,'%s,%f,%f\n',ships_ratio_name(i),ships_ratio_tonnage(i),ships_ratio_width(i));
30 end
31 -
32 fclose(fid);
```

Task 2, 3 and4)

```
Command Window
>> ships_tonnage200000_number
ships_tonnage200000_number =
7

>> ships_tonnage200000_name
ships_tonnage200000_name =
7×1 cell array

{'Wonder Of The Seas' }
{'Symphony Of The Seas' }
{'Harmony Of The Seas' }
{'Allure Of The Seas' }
{'Oasis Of The Seas' }
{'Global Dream' }
{'MSC Europa' }

>> ships_length325_number
ships_length325_number =
7

>> ships_length325_name
ships_length325_name =
7×1 cell array

{'MSC Meraviglia' }
{'MSC Bellissima' }
{'MSC Seaside' }
{'MSC Seaview' }
{'Navigator Of The Seas'}
{'Mariner Of The Seas' }
{'Explorer Of The Seas' }
fx
```

```
Command Window
>> ship_width_average
ship_width_average =
44.9818

>> ships_ratio_name
ships_ratio_name =
43×1 cell array

{'MSC Europa' }
{'Arvia' }
{'Iona' }
{'AIDAcosma' }
{'AIDAnova' }
{'Coste Smeralda' }
{'Coste Toscana' }
{'Carnival Celebration' }
{'Carnival Mardi Gras' }
{'MSC Grandiosa' }
{'MSC Virtuosa' }
{'MSC Meraviglia' }
{'MSC Bellissima' }
{'MSC Seashore' }
{'Odyssey Of The Seas' }
{'Norwegian Bliss' }
{'Norwegian Encore' }
{'Norwegian Joy' }
{'Norwegian Escape' }
{'Norwegian Epic' }
{'Freedom Of The Seas' }
{'Independence Of The Seas' }
{'Liberty Of The Seas' }
{'MSC Seaside' }
{'MSC Seaview' }
{'Queen Mary 2' }
{'Norwegian Getaway' }
fx
```

```

Command Window
{'Costa Toscana'      }
{'Carnival Celebration' }
{'Carnival Mardi Gras' }
{'MSC Grandiosa'       }
{'MSC Virtuosa'        }
{'MSC Meraviglia'      }
{'MSC Bellissima'       }
{'MSC Seashore'         }
{'Odyssey Of The Seas' }
{'Norwegian Bliss'      }
{'Norwegian Encore'     }
{'Norwegian Joy'        }
{'Norwegian Escape'     }
{'Norwegian Epic'       }
{'Freedom Of The Seas'  }
{'Independence Of The Seas'}
{'Liberty Of The Seas'  }
{'MSC Seaside'          }
{'MSC Seaview'          }
{'Queen Mary 2'          }
{'Norwegian Getaway'    }
{'Norwegian Breakaway'  }
{'Disney Wish'          }
{'Britannia'             }
{'Discovery Princess'   }
{'Enchanted Princess'   }
{'Sky Princess'          }
{'Majestic Princess'    }
{'Regal Princess'        }
{'Royal Princess'        }
{'Navigator Of The Seas' }
{'MSC Divina'            }
{'MSC Preziosa'          }
{'Mariner Of The Seas'  }
{'MSC Fantasia'          }
{'MSC Splendida'         }
{'Explorer Of The Seas'  }

```

Task 5)

Editor - G:\My Drive\Semester 2\Computer Application in CEE-
 Assignment5_Problem3.m | Cruise_ships.m | ships:

```

1 Ship Name,Tonnage,Width(meter)
2 MSC Europa,205700,47
3 Arvia,184700,42
4 Iona,184700,42
5 AIDAcosma,183900,42
6 AIDAnova,183900,42
7 Costa Smeralda,183900,42
8 Costa Toscana,183900,42
9 Carnival Celebration,183900,42
10 Carnival Mardi Gras,183900,42
11 MSC Grandiosa,177100,43
12 MSC Virtuosa,177100,43
13 MSC Meraviglia,171598,43
14 MSC Bellissima,171598,43
15 MSC Seashore,169380,41
16 Odyssey Of The Seas,169300,49
17 Norwegian Bliss,168028,42
18 Norwegian Encore,167800,42
19 Norwegian Joy,167400,42
20 Norwegian Escape,163000,42
21 Norwegian Epic,155873,40
22 Freedom Of The Seas,154407,56
23 Independence Of The Seas,154407,56
24 Liberty Of The Seas,154407,56
25 MSC Seaside,152050,43
26 MSC Seaview,152050,43
27 Queen Mary 2,149528,44
28 Norwegian Getaway,146600,51
29 Norwegian Breakaway,145655,51
30 Disney Wish,144000,41

```

Command Window

f1 >>