
```
% -----  
% Name: Clancy Crawford  
% Section: EGR115 - Section 12  
% Submission Date: 12/6/2024  
%  
% File Description: This program takes the values from the Fitness  
% Tracker Data spreadsheet and adds, removes, filters, views, and  
% search/views data.  
%  
% Citation: [  
%             Resources | Line Number  
%             Creation of Excel Data: https://chatgpt.com/ | 77-end  
%         ]  
% -----  
  
clear  
clc  
close all  
  
% INTRODUCTION  
fprintf('Welcome. This program''s functions relate to the Fitness Tracker  
Data spreadsheet.\n')  
fprintf('Functions Include:\n\tAdd Data\n\tRemove Data (should view data  
before choosing this option to find what you want to remove)\n\tFilter Data  
(only numerical/date data)\n\tView Data\n\tSearch/View Data\n');  
  
%initializing variables  
repeat = 'yes';  
view_set = 1;  
  
while strcmpi(repeat, 'yes')  
    %initializing changing variables  
    counter = 0;  
    row_search = 1;  
  
    %initializing user input and error checking for what the user wants to  
    %do and which sheet they want to do it in  
    rand_check = input('Do you want the fucntion to be chosen at random? ',  
's');  
  
    rand_check = error_check_yes_no(rand_check);  
  
    if strcmpi(rand_check, 'yes')  
        rand_val = randi(5); %<SM:RANDOM:CRAWFORD>  
  
        %assigning variables to the random values and getting intention  
        if rand_val == 1  
            intention = 'add data';
```

```

        elseif rand_val == 2
            intention = 'remove data';
        elseif rand_val == 3
            intention = 'filter data';
        elseif rand_val == 4
            intention = 'view data';
        elseif rand_val == 5
            intention = 'search/view data';
    end

    fprintf('This program is going to %s.\n', intention);
else
    intention = input('What do you want this program to do? Please input
a non-plural function. ', 's');
    while isempty(intention) || strcmpi(intention, 'Add Data') ~= 1 &&
strcmpi(intention, 'Remove Data') ~= 1 && strcmpi(intention, 'Filter Data')
~= 1 && strcmpi(intention, 'View Data') ~= 1 && strcmpi(intention, 'Search/
View Data') ~= 1 %<SM:STRING:CRAWFORD>
        intention = input('Error. What do you want this program to do?
', 's');
    end
end

fprintf('There are different data sets in the fitness
tracker. They Include: \n\tWorkout Sessions\n\tDiet Plans\n\tBody
Measurements\n\tGoals\n\tInsights\n');
worksheet = input('What sheet do you want to work in? ', 's');

while isempty(worksheet) || strcmpi(worksheet, 'Workout Sessions') ~= 1 &&
strcmpi(worksheet, 'Diet Plans') ~= 1 && strcmpi(worksheet,
'Data Measurements') ~= 1 && strcmpi(worksheet, 'Goals') ~= 1 &&
strcmpi(worksheet, 'Insights') ~= 1
    worksheet = input('Error. What sheet would you like to choose? ',
's');
end

%initializing from the excel sheet and a new array to store search
%values in
fitness_data = readcell('Fitness_Tracker_Data.xlsx','Sheet', worksheet);
[n_rows, n_cols] = size(fitness_data); %<SM:READ:CRAWFORD>
search_result = cell(n_rows, n_cols);

%ADD DATA
if strcmpi(intention, 'Add Data') == 1
    fprintf('Type the data you want to add when prompted and close the
EXCEL sheet if it''s open.\n');

    fitness_data = readcell('Fitness_Tracker_Data.xlsx','Sheet',
worksheet);
    [n_rows, n_cols] = size(fitness_data);

    %workout sessions sheet
    if strcmpi(worksheet, 'Workout Sessions') == 1

```

```

% - error checking input values
date = input('Date (MM/DD/YYYY): ', 's');
while isempty(date) || isstring(date)
    date = input('Error. Enter an date.', 's');
end

excersize = input('Exercise: ', 's');
while isempty(excersize) || isstring(excersize)
    excersize = input('Error. Enter an excersize.', 's');
end

category = input('Category of Exercise: ', 's');
while isempty(category) || isstring(category)
    category = input('Error. Enter a category.', 's');
end

duration = input('Duration (min): ');
while isempty(duration) || duration < 0 || mod(duration, 1) ~= 0
    duration = input('Error. Enter a duration (min).', 's');
end

cals = input('Calories Burned: ');
while isempty(cals) || cals < 0 || mod(cals, 1) ~= 0
    cals = input('Error. Enter a calorie amount.', 's');
end

notes = input('Notes: ', 's');
while isempty(notes) || isstring(notes)
    notes = input('Error. Enter a note.', 's');
end

% adding data to the sheet
new_row = {date, excersize, category, duration, cals, notes};
fitness_data(end + 1, :) = new_row;

%diet plans sheet
elseif strcmpi(worksheet, 'Diet Plans') == 1

% - error checking input values
date = input('Date (MM/DD/YYYY): ', 's');
while isempty(date) || isstring(date)
    date = input('Error. Enter an date.', 's');
end

meal_type = input('Meal Type (Ex: Breakfast, Lunch, Snack, etc.): ', 's');
while isempty(meal_type) || isstring(meal_type)
    meal_type = input('Error. Enter a meal type.', 's');
end

food_item = input('Food Item: ', 's');
while isempty(food_item) || isstring(food_item)
    food_item = input('Error. Enter a food item.', 's');
end

```

```
protien = input('Protien (g): ');
while isempty(protien) || protien < 0 || mod(protien, 1) ~= 0
    protien = input('Error. Enter a protein amount.');
end

carbs = input('Carbs (g): ');
while isempty(carbs) || carbs < 0 || mod(carbs, 1) ~= 0
    carbs = input('Error. Enter a carb amount.');
end

fat = input('Fat (g): ');
while isempty(fat) || fat < 0 || mod(fat, 1) ~= 0
    fat = input('Error. Enter a fat amount.');
end

cals = input('Calories: ');
while isempty(cals) || cals < 0 || mod(cals, 1) ~= 0
    cals = input('Error. Enter a calorie amount.');
end

notes = input('Notes: ', 's');
while isempty(notes) || isstring(notes)
    notes = input('Error. Enter a note.', 's');
end

% adding data to the sheet
new_row = {date, meal_type, food_item, cals, protien, carbs,
fat, notes};
fitness_data(end + 1, :) = new_row;

%body measurements sheet
elseif strcmpi(worksheet, 'Body Measurements') == 1

% - error checking input values
date = input('Date (MM/DD/YYYY): ', 's');
while isempty(date) || isstring(date)
    date = input('Error. Enter an date.', 's');
end

weight = input('Weight (lbs): ');
while isempty(weight) || weight < 0
    weight = input('Error. Enter an weight.');
end

body_fat = input('Body Fat (%): ');
while isempty(body_fat) || body_fat < 0
    body_fat = input('Error. Enter a body fat percentage.');
end

waist = input('Waist (in): ');
while isempty(waist) || waist < 0
    waist = input('Error. Enter a waist measurement.', 's');
end
```

```
chest = input('Chest (in): ');
while isempty(chest) || chest < 0
    chest = input('Error. Enter a chest measurement.');
end

notes = input('Notes: ', 's');
while isempty(notes) || isstring(notes)
    notes = input('Error. Enter a note.', 's');
end

% adding data to the sheet
new_row = {date, weight, body_fat, waist, chest, notes};
fitness_data(end + 1, :) = new_row;

%goals sheet
elseif strcmpi(worksheet, 'Goals') == 1

% - error checking input values
goal_type = input('Goal Type: ', 's');
while isempty(goal_type) || isstring(goal_type)
    goal_type = input('Error. Enter goal type.', 's');
end

goal_des = input('Goal Description: ', 's');
while isempty(goal_des) || isstring(goal_des)
    goal_des = input('Error. Enter goal description.', 's');
end

target_val = input('Target Value: ', 's');
while isempty(target_val) || isstring(target_val)
    target_val = input('Error. Enter target value.', 's');
end

current_val = input('Current Value: ', 's');
while isempty(current_val) || isstring(current_val)
    current_val = input('Error. Enter current value.', 's');
end

deadline = input('Deadline (MM/DD/YYYY): ', 's');
while isempty(deadline) || isstring(deadline)
    deadline = input('Error. Enter a deadline.', 's');
end

notes = input('Notes: ', 's');
while isempty(notes) || isstring(notes)
    notes = input('Error. Enter a note.', 's');
end

% adding data to the sheet
new_row = {goal_type, goal_des, target_val, current_val,
deadline, notes};
fitness_data(end + 1, :) = new_row;
```

```

        %insights sheet
    elseif strcmpi(worksheet, 'Insights') == 1

        % - error checking input values
        metric = input('Metric (Ex: Fastest 5K Time, Most Calories
Burned in a Session, etc): ', 's');
        while isempty(metric) || isstring(metric)
            metric = input('Error. Enter metric.', 's');
        end

        best_val = input('Best Value: ', 's');
        while isempty(best_val) || isstring(best_val)
            best_val = input('Error. Enter the best value.', 's');
        end

        current_val = input('Current Value: ', 's');
        while isempty(current_val) || isstring(current_val)
            current_val = input('Error. Enter current value.', 's');
        end

        notes = input('Notes: ', 's');
        while isempty(notes) || isstring(notes)
            notes = input('Error. Enter a note.', 's');
        end

        % adding data to the sheet
        new_row = {metric, best_val, current_val, notes};
        fitness_data(end + 1, :) = new_row;
    end

    %saving the sheet
    writecell(fitness_data, 'Fitness_Tracker_Data.xlsx', 'Sheet',
    worksheet)

    fprintf('The data is now added to the spreadsheet.\n');

    %REMOVE DATA
    elseif strcmpi(intention, 'Remove Data') == 1

        fitness_data = readcell('Fitness_Tracker_Data.xlsx', 'Sheet',
        worksheet);
        [n_rows, n_cols] = size(fitness_data);

        %input error checking for what row the user wants to remove
        remove_row = input('What row would you like to remove? (Integer
higher than 1, where 1 = heading and 2 = 1st row) ');
        while isempty(remove_row) || remove_row < 2 || remove_row > n_rows
        || mod(remove_row, 1) ~= 0
            remove_row = input('Error. Please, enter a number for which row
you want to remove. ');
        end

        % this code works VV

```

```

    disp(fitness_data)
    for j = 1:width(fitness_data)
        fitness_data{remove_row, j} = [];
    end
    disp(fitness_data)
    writecell(fitness_data, 'Fitness_Tracker_Data.xlsx', 'Sheet',
    worksheet);

    % this code works ^^

    % you said don't worry about this because I did the code right and
    it works,
    % excel and matlab are just not working how I need
    disp(fitness_data)
    fitness_data_new = readcell('Fitness_Tracker_Data.xlsx', 'Sheet',
    worksheet);
    disp(fitness_data_new)

    fprintf('The data is now removed from the spreadsheet.\n');

    %FILTER DATA
    elseif strcmpi(intention, 'Filter Data') == 1

        fitness_data = readcell('Fitness_Tracker_Data.xlsx', 'Sheet',
    worksheet);
        [n_rows, n_cols] = size(fitness_data);

        %WORKOUT SESSIONS SHEET
        if strcmpi(worksheet, 'Workout Sessions') == 1

            %columns for the sheet
            fprintf('\n\tDate (MM/DD/YYYY) = 1 \n\tDuration (min) = 4
\n\tCalories Burned = 5\n')

            %initializing and error checking what column the user wants to
            %search in
            col_search = input('What column would you like to search? ');

            while isempty(col_search) || col_search < 0 || col_search > 5 ||
mod(col_search, 1) ~= 0 %<SM:WHILE:CRAWFORD> %<SM:BUILT-FUNC:CRAWFORD>
                col_search = input('Error. Please, enter a column number: ');
            end

            % DIET PLANS SHEET
            elseif strcmpi(worksheet, 'Diet Plans') == 1

                %columns for the sheet
                fprintf('Columns you can filter include:\n\tDate (MM/DD/YYYY) =
1 \n\tCalories = 4 \n\tProtein = 5 \n\tCarbs = 6 \n\tFat = 7\n')

                %initializing and error checking what column the user wants to
                %search in
                col_search = input('What column would you like to search? ');

```

```

        while isempty(col_search) || col_search < 0 || col_search > 7 ||
mod(col_search, 1) ~= 0
            col_search = input('Error. Please, enter a column number: ');
end

% BODY MEASUREMENT SHEET
elseif strcmpi(worksheet, 'Body Measurements') == 1

    %columns for the sheet
    fprintf('Columns you can filter include:\n\tDate (MM/DD/YYYY) =
1\n\tWeight (lbs) = 2\n\tBody Fat (%) = 3\n\tWaist (in) = 4\n\tChest
(in)= 5\n');

    %initializing and error checking what column the user wants to
%search in
    col_search = input('What column would you like to search? ');

    while isempty(col_search) || col_search < 0 || col_search > 5 ||
mod(col_search, 1) ~= 0
            col_search = input('Error. Please, enter a column number: ');
end

% GOALS SHEET
elseif strcmpi(worksheet, 'Goals') == 1

    fprintf('You can filter the Deadlines in this sheet.\n')
    col_search = 5;

% INSIGHTS SHEET
elseif strcmpi(worksheet, 'Insights') == 1

    fprintf('There are only words present in this sheet so there is
no way to filter them');

end

%getting filtering threshold for a datetime value and converting to
datetime
if strcmpi(worksheet, 'Goals') == 1 || strcmpi(worksheet, 'Body
Measurements') == 1 && col_search == 1 || strcmpi(worksheet, 'Diet Plans')
== 1 && col_search == 1 || strcmpi(worksheet, 'Workout Sessions') == 1 &&
col_search == 1
    filter = input('Enter the date threshold you want to
filter(MM/dd/yyyy): ', 's');
    filter = datetime(filter, 'InputFormat', 'MM/dd/yyyy');

    %getting filtering threshold for a numerical value
else
    filter = input('Enter the threshold for filtering (numerical):
');

while isempty(filter) || filter < 0
    filter = input('Error. Enter a threshold for filtering. ');

```

```

        end
    end

    filtered_result = [];

    %filtering
    for k = 2: n_rows
        %filtering if a number
        if isnumeric(filter)
            if filter > fitness_data{k, col_search}
                filtered_result = [filtered_result; fitness_data(k, :)];
            end
        %filtering if a date
        else
            if filter > datetime(fitness_data{k, col_search},
'InputFormat', 'MM/dd/yyyy')
                filtered_result = [filtered_result; fitness_data(k, :)];
            end
        end
    end

    end

    %presenting filtered data
    % workout sessions sheet
    if strcmpi(worksheets, 'Workout Sessions') == 1

        date = datetime(filtered_result(2:end, 1), 'InputFormat', 'MM/dd/
yyyy');
        duration = cell2mat(filtered_result(2:end, 4));
        calories = cell2mat(filtered_result(2:end, 5));
        notes = char(filtered_result(2:end, 6));
        exercise_rows = char(filtered_result(2:end, 2));
        category_rows = char(filtered_result(2:end, 3));

        title_labels = {'Date', 'Exercise', 'Category', 'Duration
(min)', 'Calories Burned', 'Notes'};
        w_s_table = table(date, exercise_rows, category_rows, duration,
calories, notes, 'VariableNames', title_labels);
        disp(w_s_table)

        %diet plans sheet
        elseif strcmpi(worksheets, 'Diet Plans') == 1

            date = datetime(filtered_result(2:end, 1), 'InputFormat', 'MM/dd/
yyyy');
            calories = cell2mat(filtered_result(2:end, 4));
            protien = cell2mat(filtered_result(2:end, 5));
            carbs = cell2mat(filtered_result(2:end, 6));
            fat = cell2mat(filtered_result(2:end, 7));
            meal_type_rows = char(filtered_result(2:end, 2));
            food_item_rows = char(filtered_result(2:end, 3));
            notes = char(filtered_result(2:end, 8));

            title_labels = {'Date', 'Meal Type', 'Food Item', 'Calories',

```

```

'Protien (g)', 'Carbs (g)', 'Fat (g)', 'Notes');
d_p_table = table(date, meal_type_rows, food_item_rows,
calories, protien, carbs, fat, notes, 'VariableNames',title_labels);
disp(d_p_table)

% body measurement sheet
elseif strcmpi(worksheet, 'Body Measurements') == 1

date = datetime(filtered_result(2:end, 1), 'InputFormat','MM/dd/
yyyy');
weight = cell2mat(filtered_result(2:end, 2));
body_fat = cell2mat(filtered_result(2:end, 3));
waist = cell2mat(filtered_result(2:end, 4));
chest = cell2mat(filtered_result(2:end, 5));
notes = char(filtered_result(2:end, 6));

title_labels = {'Date', 'Weight', 'Body Fat', 'Waist (in)', 'Chest (in)', 'Notes'};
b_m_table = table(date, weight, body_fat, waist, chest, notes, 'VariableNames',title_labels);
disp(b_m_table)

% goals sheet
elseif strcmpi(worksheet, 'Goals') == 1

goal_type = char(filtered_result(2:end, 1));
goal_description = char(string(filtered_result(2:end, 2)));
target_val = char(string(filtered_result(2:end, 3)));
current_val = char(string(filtered_result(2:end, 4)));
deadline = datetime(filtered_result(2:end, 5));
notes = char(filtered_result(2:end, 6));

title_labels = {'Goal Type', 'Goal Description', 'Target Value', 'Current Value', 'Deadline,' , 'Notes'};
b_m_table = table(goal_type, goal_description, target_val, current_val, deadline, notes, 'VariableNames',title_labels);
disp(b_m_table)
end

%VIEW DATA
elseif strcmpi(intention, 'View Data') == 1

fitness_data = readcell('Fitness_Tracker_Data.xlsx', 'Sheet',
worksheet);
[n_rows, n_cols] = size(fitness_data);

%WORKOUT SESSIONS SHEET
if strcmpi(worksheet, 'Workout Sessions') == 1

% converting the columns to a format matlab can plot
date = datetime(fitness_data(2:end, 1), 'InputFormat','MM/dd/
yyyy');
duration = cell2mat(fitness_data(2:end, 4));
calories = cell2mat(fitness_data(2:end, 5));

```

```

notes = char(fitness_data(2:end, 6));
exercise_rows = char(fitness_data(2:end, 2));
category_rows = char(fitness_data(2:end, 3));

%finding frequency of exercise
exercise = rot90(fitness_data(2:end, 2));
n_of_exercises = unique(exercise);
c_exercise = zeros(size(n_of_exercises));
for i = 1:length(n_of_exercises)
    c_exercise(i) = sum(strcmpi(exercise, n_of_exercises{i}));
end

%finding frequency of category
category = rot90(fitness_data(2:end, 3));
n_of_category = unique(category);
c_category = zeros(size(n_of_category));
for i = 1: length(n_of_category)
    c_category(i) = sum(strcmpi(category, n_of_category{i}));
end

% Exercise frequency
figure('WindowState', 'maximized')
subplot(4, 1, 1)
bar(n_of_exercises, c_exercise, 'r')
title('Workout Sessions: Exercise Frequency')
xlabel("Exercise")
ylabel('Frequency')
axis padded
% Category frequency
subplot(4, 1, 2)
bar(n_of_category, c_category, 'b')
title('Workout Sessions: Category Frequency')
xlabel("Category")
ylabel('Frequency')
axis padded
% duration per workout
subplot(4, 1, 3)
plot(date, duration, 'm') %<SM: PLOT:CRAWFORD>
title('Workout Sessions: Duration per Workout')
xlabel("Date")
ylabel('Duration (minutes)')
% calories per workout
subplot(4, 1, 4)
plot(date, calories , 'g')
title('Workout Sessions: Calories per Workout')
xlabel("Date")
ylabel('Calories')

%displaying a table
display_table = input('Would you like to display the sheet? ',
's'); %<SM: PDF_PARAM:CRAWFORD>

display_table = error_check_yes_no(display_table);

```

```

if strcmpi(display_table, 'yes') == 1 %<SM:PDF_RETURN:CRAWFORD>
    title_labels = {'Date', 'Exercise', 'Category', 'Duration
(min)', 'Calories Burned', 'Notes'};
    w_s_table = table(date, exercise_rows, category_rows,
duration, calories, notes, 'VariableNames',title_labels);
    disp(w_s_table)
end

% DIET PLANS SHEET
elseif strcmpi(worksheet, 'Diet Plans') == 1

% converting the columns to a format matlab can plot
date = datetime(fitness_data(2:end, 1), 'InputFormat','MM/dd/
yyyy');
each_date = unique(date);
calories = cell2mat(fitness_data(2:end, 4));
protien = cell2mat(fitness_data(2:end, 5));
carbs = cell2mat(fitness_data(2:end, 6));
fat = cell2mat(fitness_data(2:end, 7));
array_p_c_f = [protien, carbs, fat];
meal_type_rows = char(fitness_data(2:end, 2));
food_item_rows = char(fitness_data(2:end, 3));
notes = char(fitness_data(2:end, 8));

%finding frequency of meal type
meal_type = rot90(fitness_data(2:end, 2));
n_of_meal_type = unique(meal_type);
c_meal_type = zeros(size(n_of_meal_type));
for i = 1:length(n_of_meal_type)
    c_meal_type(i) = sum(strcmpi(meal_type, n_of_meal_type{i}));
end

% Meal Type frequency
figure()
subplot(3, 1, 1)
bar(n_of_meal_type, c_meal_type, 'r')
title('Diet Plans: Meal Type Frequency')
xlabel("Meal Type")
ylabel('Frequency')
axis padded
% Calories per meal
subplot(3, 1, 2)
plot(date, calories, 'm')
title('Diet Plans: Calories per Meal')
xlabel("Date")
ylabel('Calories')
% Protien, carbs, and fat vs Date & Meal Types
subplot(3, 1, 3)
bar(array_p_c_f)
title('Diet Plans: Protien, Carbs, and Fat vs Date & Meal Types')
xlabel('Date and Meal Plans (1 = 11/1/24 Breakfast, etc.)')
ylabel('Grams')
legend('Protien', 'Carbs', 'Fat')
axis padded

```

```

    %displaying a table
    display_table = input('Would you like to display the sheet? ',
's');

    display_table = error_check_yes_no(display_table);

    if strcmpi(display_table, 'yes') == 1
        title_labels = {'Date', 'Meal Type', 'Food Item', 'Calories',
'Protien (g)', 'Carbs (g)', 'Fat (g)', 'Notes'};
        d_p_table = table(date, meal_type_rows, food_item_rows,
calories, protien, carbs, fat, notes, 'VariableNames',title_labels);
        disp(d_p_table)
    end

    % BODY MEASUREMENT SHEET
    elseif strcmpi(worksheet, 'Body Measurements') == 1

        % converting the columns to a format matlab can plot
        date = datetime(fitness_data(2:end, 1), 'InputFormat', 'MM/dd/
yyyy');

        weight = cell2mat(fitness_data(2:end, 2));
        body_fat = cell2mat(fitness_data(2:end, 3));
        waist = cell2mat(fitness_data(2:end, 4));
        chest = cell2mat(fitness_data(2:end, 5));
        notes = char(fitness_data(2:end, 6));

        % Weight fluctuation
        figure()
        subplot(2, 1, 1)
        plot(date, weight, 'p')
        title('Body Measurements: Weight Fluctuation')
        xlabel("Date")
        ylabel('Weight (lbs)')
        axis padded
        % body fat and waist measurement vs date
        subplot(2, 1, 2)
        bar(date, [body_fat, waist, chest])
        title('Body Measurement: Body Fat, Waist, and Chest Measurements
vs Date')
        xlabel('Date')
        ylabel('Measurements')
        legend('Body Fat (%)', 'Waist (in)', 'Chest (in)')

        %displaying a table
        display_table = input('Would you like to display the sheet? ',
's');

        display_table = error_check_yes_no(display_table);

        if strcmpi(display_table, 'yes') == 1
            title_labels = {'Date', 'Weight', 'Body Fat', 'Waist (in)',
'Chest (in)', 'Notes'};
            b_m_table = table(date, weight, body_fat, waist, chest,
notes, 'VariableNames', title_labels);
            disp(b_m_table)
        end
    end

```

```

notes, 'VariableNames', title_labels);
    disp(b_m_table)
end

% GOALS SHEET
elseif strcmpi(worksheet, 'Goals') == 1

    % converting the columns to a format matlab can plot
goal_type = char(fitness_data(2:end, 1));
goal_description = char(string(fitness_data(2:end, 2)));
target_val = char(string(fitness_data(2:end, 3)));
current_val = char(string(fitness_data(2:end, 4)));
deadline = datetime(fitness_data(2:end, 5));
notes = char(fitness_data(2:end, 6));

fprintf('There are no values to compare in this worksheet.\n')

%displaying a table
display_table = input('Would you like to display the sheet? ',
's');

display_table = error_check_yes_no(display_table);

if strcmpi(display_table, 'yes') == 1
    title_labels = {'Goal Type', 'Goal Description', 'Target
Value', 'Current Value', 'Deadline', 'Notes'};
    b_m_table = table(goal_type, goal_description, target_val,
current_val, deadline, notes, 'VariableNames', title_labels);
    disp(b_m_table)
end

%INSIGHTS SHEET
elseif strcmpi(worksheet, 'Insights') == 1

    % converting the columns to a format matlab can plot
metric = char(fitness_data(2:end, 1));
best_val = char(string(fitness_data(2:end, 2)));
current_val = char(string(fitness_data(2:end, 3)));
notes = char(fitness_data(2:end, 4));

fprintf('There are no values to compare in this worksheet.\n')

%displaying a table
display_table = input('Would you like to display the sheet? ',
's');

display_table = error_check_yes_no(display_table);

if strcmpi(display_table, 'yes') == 1
    title_labels = {'Metric', 'Best Value', 'Current Value',
'Notes'};
    b_m_table = table(metric, best_val, current_val, notes,
'VariableNames', title_labels);
    disp(b_m_table)

```

```

    end

    end

    %SEARCH AND VIEW DATA
    elseif strcmpi(intention, 'Search/View Data') == 1

        fitness_data = readcell('Fitness_Tracker_Data.xlsx', 'Sheet',
worksheet);
        [n_rows, n_cols] = size(fitness_data);

        %WORKOUT SESSIONS SHEET
        if strcmpi(worksheet, 'Workout Sessions') == 1

            %columns for the sheet
            fprintf('\n\tDate (MM/D/YYYY) = 1 \n\tExercise = 2\n\tCategory =
3 \n\tDuration (min) = 4 \n\tCalories Burned = 5 \n\tNotes = 6\n')

            %initializing and error checking what column the user wants to
            %search in
            col_search = input('What column would you like to search? ');

            while isempty(col_search) || col_search < 0 || col_search > 6 ||
mod(col_search, 1) ~= 0 %<SM:WHILE:CRAWFORD> %<SM:BUILT-FUNC:CRAWFORD>
                col_search = input('Error. Please, enter a column number: ');
            end

            % separating number value columns and string columns

            %number columns
            if col_search == 4 || col_search == 5 %<SM:ROP:CRAWFORD>

                %initializing and error checking search value
                search_n = input('Please, enter a number you want to search
and view for: ');

                while isempty(search_n) || mod(search_n, 1) ~=0 || search_n
< 0 %<SM:BOP:CRAWFORD>
                    search_n = input('Error. Enter a number you want to
search for');
                end

                % putting values that meet search criteria into a new array
                for k = 2: n_rows %<SM:FOR:CRAWFORD> %<SM:SEARCH:CRAWFORD>
                    if search_n == fitness_data{k, col_search}
%<SM:IF:CRAWFORD>
                        for j = 1: n_cols
                            search_result(row_search, j) = fitness_data(k,
j);
                        end
                        row_search = row_search + 1;
                        counter = counter + 1;
                    end
                end
            end
        end
    end

```

```

        %string columns
    else

        %initializing and error checking search value
        search_s = input('Please, enter a string you want to search
and view for: ', 's');
        while isempty(search_s) || isstring(search_s)
            search_s = input('Error. Enter a string you want to
search for', 's');
        end

        % putting values that meet search criteria into a new array
        for k = 2: n_rows
            if strcmpi(search_s, fitness_data{k, col_search}) == 1
                for j = 1: n_cols
                    search_result(row_search, j) = fitness_data(k,
j);
                end
                row_search = row_search + 1;
                counter = counter + 1;
            end
        end
    end

    %DIET PLANS SHEET
    elseif strcmpi(worksheet, 'Diet Plans') == 1

        %columns for the sheet
        fprintf('Columns Include:\n\tDate (MM/D/YYYY) = 1 \n\tMeal Type
= 2 \n\tFood Item = 3 \n\tCalories = 4 \n\tProtien = 5 \n\tCarbs = 6 \n\tFat
= 7 \n\tNotes = 8\n');

        %initializing and error checking what column the user wants to
        %search in
        col_search = input('What column would you like to search? ');

        while isempty(col_search) || col_search < 0 || col_search > 8 ||
mod(col_search, 1) ~= 0
            col_search = input('Error. Please, enter a column number: ');
        end

        % seperating number value columns and string columns

        %number columns
        if col_search == 4 || col_search == 5 || col_search == 6 ||
col_search == 7

            %initializing and error checking search value
            search_n = input('Please, enter a number you want to search
and view for: ');

            while isempty(search_n) || mod(search_n, 1) ~=0 || search_n

```

```

< 0
    search_n = input('Error. Enter a number you want to
search for');
    end

    % putting values that meet search criteria into a new array
    for k = 2: n_rows
        if search_n == fitness_data{k, col_search}
            for j = 1: n_cols
                search_result(row_search, j) = fitness_data(k,
j);
            end
            row_search = row_search + 1;
            counter = counter + 1;
        end
    end

    %string columns
else

    %initializing and error checking search value
    search_s = input('Please, enter a string you want to search
and view for: ', 's');
    while isempty(search_s) || issstring(search_s)
        search_s = input('Error. Enter a string you want to
search for', 's');
    end

    % putting values that meet search criteria into a new array
    for k = 2: n_rows
        if strcmpi(search_s, fitness_data{k, col_search}) == 1
            for j = 1: n_cols
                search_result(row_search, j) = fitness_data(k,
j);
            end
            row_search = row_search + 1;
            counter = counter + 1;
        end
    end
end

%BODY MEASUREMENTS SHEET
elseif strcmpi(worksheet, 'Body Measurements') == 1

    %columns for the sheet
    fprintf('Columns Include:\n\tDate (MM/D/YYYY) = 1 \n\tWeight
(lbs) = 2 \n\tBody Fat (%) = 3 \n\tWaist (in) = 4 \n\tChest (in)= 5
\n\tNotes = 6\n');

    %initializing and error checking what column the user wants to
    %search in
    col_search = input('What column would you like to search? ');

```

```

        while isempty(col_search) || col_search < 0 || col_search > 6 ||
mod(col_search, 1) ~= 0
            col_search = input('Error. Please, enter a column number: ');
        end

        % separating number value columns and string columns

        %number columns
        if col_search == 2 || col_search == 3 || col_search == 4 ||
col_search == 5

            %initializing and error checking search value
            search_n = input('Please, enter a number you want to search
and view for: ');

            while isempty(search_n) || search_n < 0
                search_n = input('Error. Enter a number you want to
search for');
            end

            % putting values that meet search criteria into a new array
            for k = 2: n_rows
                if search_n == fitness_data{k, col_search}
                    for j = 1: n_cols
                        search_result(row_search, j) = fitness_data(k,
j);
                    end
                    row_search = row_search + 1;
                    counter = counter + 1;
                end
            end
        end

        %string columns
    else

        %initializing and error checking search value
        search_s = input('Please, enter a string you want to search
and view for: ', 's');
        while isempty(search_s) || issstring(search_s)
            search_s = input('Error. Enter a string you want to
search for', 's');
        end

        % putting values that meet search criteria into a new array
        for k = 2: n_rows
            if strcmpi(search_s, fitness_data{k, col_search}) == 1
                for j = 1: n_cols
                    search_result(row_search, j) = fitness_data(k,
j);
                end
                row_search = row_search + 1;
                counter = counter + 1;
            end
        end
    end

```

```

    end

    %GOALS SHEET
    elseif strcmpi(worksheet, 'Goals') == 1

        %columns for the sheet
        fprintf('Columns Include:\n\tGoal Type = 1 \n\tGoal Description
= 2 \n\tTarget Value = 3 \n\tCurrent Value = 4 \n\tDeadline (MM/D/YYYY) = 5
\n\tNotes = 6\n')

        %initializing and error checking what column the user wants to
        %search in
        col_search = input('What column would you like to search? ');

        while isempty(col_search) || col_search < 0 || col_search > 6 ||
mod(col_search, 1) ~= 0
            col_search = input('Error. Please, enter a column number: ');
        end

        %string columns only for this sheet
        %initializing and error checking search value
        search_s = input('Please, enter a string you want to search and
view for: ', 's');
        while isempty(search_s) || isstring(search_s)
            search_s = input('Error. Enter a string you want to search
for', 's');
        end

        % putting values that meet search critera into a new array
        for k = 2: n_rows
            if strcmpi(search_s, fitness_data{k, col_search}) == 1
                for j = 1: n_cols
                    search_result(row_search, j) = fitness_data(k, j);
                end
                row_search = row_search + 1;
                counter = counter + 1;
            end
        end

        %INSIGHTS SHEET
    elseif strcmpi(worksheet, 'Insights') == 1

        %columns for the sheet
        fprintf('Columns Include:\n\tMetric = 1 \n\tBest Value = 2
\n\tCurrent Value = 3 \n\tNotes = 4\n')

        %initializing and error checking what column the user wants to
        %search in
        col_search = input('What column would you like to search? ');

        while isempty(col_search) || col_search < 0 || col_search > 4 ||
mod(col_search, 1) ~= 0
            col_search = input('Error. Please, enter a column number: ');
        end

```

```

    end

    %string columns only for this sheet
    %initializing and error checking search value
    search_s = input('Please, enter a string you want to search and
view for: ', 's');
        while isempty(search_s) || isstring(search_s)
            search_s = input('Error. Enter a string you want to search
for', 's');
        end

        % putting values that meet search critera into a new array
        for k = 2: n_rows
            if strcmpi(search_s, fitness_data{k, col_search}) == 1
%<SM:REF:CRAWFORD>
                for j = 1: n_cols
                    search_result(row_search, j) = fitness_data(k, j);
%<SM:AUG:CRAWFORD>
                end
                row_search = row_search + 1;
                counter = counter + 1;
            end
        end
    end

    %checking if there is more than 1 set of data that the user is
searching for
    if counter > 1
        fprintf('\nThere are %d interations of this set of data.\n',
counter);

        %finding which set of data the user wants and error checking
        view_set = input('Which set would you like to view? ');

        while isempty(view_set) || view_set < 0 || mod(view_set, 1) ~= 0
            view_set = input('Error. Which set would you like to view? ');
        end

    end

    %displaying data
    if counter == 0
        fprintf('Data not found\n');

        %workout sessions sheet
    elseif strcmpi(worksheet, 'Workout Sessions') == 1
        fprintf('Date: %s\n', search_result{view_set, 1});
%<SM:VIEW:CRAWFORD>
        fprintf('Exercise: %s\n', search_result{view_set, 2});
        fprintf('Category: %s\n', search_result{view_set, 3});
        fprintf('Duration (min): %d\n', search_result{view_set, 4});
        fprintf('Calories Burned: %d\n', search_result{view_set, 5});
        fprintf('Notes: %s\n', search_result{view_set, 6});

```

```

        %diet plans sheet
    elseif strcmpi(worksheet, 'Diet Plans') == 1
        fprintf('Date: %s\n', search_result{view_set, 1});
        fprintf('Meal Type: %s\n', search_result{view_set, 2});
        fprintf('Food Item: %s\n', search_result{view_set, 3});
        fprintf('Calories (min): %d\n', search_result{view_set, 4});
        fprintf('Protien (g): %d\n', search_result{view_set, 5});
        fprintf('Carbs (g): %d\n', search_result{view_set, 6});
        fprintf('Fat (g): %d\n', search_result{view_set, 7});
        fprintf('Notes: %s\n', search_result{view_set, 8});

        %body measurements sheet
    elseif strcmpi(worksheet, 'Body Measurements') == 1
        fprintf('Date: %s\n', search_result{view_set, 1});
        fprintf('Weight (lbs): %0.1f\n', search_result{view_set, 2});
        fprintf('Body Fat (%): %0.1f\n', search_result{view_set, 3});
        fprintf('Waist (in): %0.1f\n', search_result{view_set, 4});
        fprintf('Chest (in): %0.1f\n', search_result{view_set, 5});
        fprintf('Notes: %s\n', search_result{view_set, 6});

        % goals sheet
    elseif strcmpi(worksheet, 'Goals') == 1
        fprintf('Goal Type: %s\n', search_result{view_set, 1});
        fprintf('Goal Description: %s\n', search_result{view_set, 2});
        fprintf('Target Value: %s\n', search_result{view_set, 3});
        fprintf('Current Value: %s\n', search_result{view_set, 4});
        fprintf('Deadline: %s\n', search_result{view_set, 5});
        fprintf('Notes: %s\n', search_result{view_set, 6});

        %insights sheet
    elseif strcmpi(worksheet, 'Insights') == 1
        fprintf('Metric: %s\n', search_result{view_set, 1});
        fprintf('Best Value: %s\n', search_result{view_set, 2});
        fprintf('Current Value: %s\n', search_result{view_set, 3});
        fprintf('Notes: %s\n', search_result{view_set, 4});
    end

end

%does the user want to repeat the code? and error checking it
repeat = input('Is there anything else you want to do to the data? (yes or no): ', 's');
repeat = error_check_yes_no(repeat);

end

%User-defined function %<SM:PDF:CRAWFORD>
function [value_checking] = error_check_yes_no(value_checking)
%ERROR_CHECK_DISPLAY_TABLE error checking yes or no input
    while isempty(value_checking) || (~strcmpi(value_checking, 'yes') &&
~strcmpi(value_checking, 'no'))
        value_checking = input('Error. Answer yes or no. ', 's');

```

```
    end
end
```

Welcome. This program's functions relate to the Fitness Tracker Data spreadsheet.

Functions Include:

Add Data

Remove Data (should view data before choosing this option to find what you want to remove)

Filter Data (only numerical/date data)

View Data

Search/View Data

Error using input

Cannot call INPUT from EVALC.

Error in CRAWFORD_12_Final_Project (line 39)

```
    rand_check = input('Do you want the fucntion to be chosen at random? ',
's');
```

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Published with MATLAB® R2024b