PTO (Paid Time Off) Calculator Report & User Manual

Programmer: Nicholas Rosato (Team 10 Section 002)

rosaton@purdue.edu

Table of Contents

Purpose/Description, Functions and Specifications……3-5

Function Diagram……6

User Manual……7-9

Employee Reference Spread Sheet……10

Sample of Data Analytics……11

Code Print-Out……12-24

Purpose/Description:

Paid time off is a huge incentive for employees at Northwest Christian Childcare (http://www.northwestchristianchildcare.org/) to come to work on time every day and work all the hours the employee is scheduled. At Northwest Christian Childcare there are issues where people are earning paid time off when they shouldn't because the system is monitored by people who are not following the rules. The way the incentive system works is that for each time an employee works the full pay cycle, the employee earns a specific amount of paid time off. Instead of having someone keep track of this and make decisions based off a spread sheet, the PTO calculator program takes over this responsibility. The way it does this is by taking in the spread sheet as a data file and breaking down information by employee. Then, it calculates whether the employee should receive paid time off and what to do if the employee is not hitting compliance hours. The program is versatile because it has options to look at in depth information based on a single employee, or calculate all the employees' PTO for the pay cycle in one "run" of the program. It is completely function based and easy to use, read, and change based on the situation.

Input: Spread sheet of employee data

Main Outputs: Employee, Straight Hours, Overtime Hours, Total Hours, Hours Scheduled, Warning, PTO Calculation, PTO Bank Status. (There are other mini inputs inside of the functions that are called, but they are irrelevant to the user).

Overview/Functions:

1. exec_function_project_rosaton
    a. Compiler function in this program; connects all functions together. This is the only function that should be ran in the code.
        i. Inputs: z
        ii. Outputs: Start, Warnings, Employee_Number
2. employee_PTO_Bank_rosaton
    a. This is the function that allows the user to adjust the code if the employee has saved up PTO from prior weeks. It is primarily used to calculate the new amount of PTO based on if the employee follows compliance.
        i. Inputs: Employee, Employee_Number, e_data, Warnings
        ii. Outputs: PTO_Bank, PTO, New_Employee, Ability
3. employee_PTO_calculator_rosaton
    a. This function calculates if a specific employee gets paid time off by meeting a set of specific criteria from Northwest Christian Childcare.
        i. Inputs: Employee, PTO_Bank, New_Employee, Ability, Warnings
        ii. Ouputs: PTO
4. employee_reference_rosaton
    a. This is a function breaks down the matrix of the numbers from the spread sheet in to employee vectors for the first group of employees in the spread sheet.
        i. Inputs: Employee_Number, e_data
        ii. Outputs: Employee

5. employee_reference_rosaton_2
    a. This is a function breaks down the matrix of the numbers from the spread sheet in to employee vectors for the first group of employees in the spread sheet.
        i. Inputs: Employee_Number, e_data
        ii. Outputs: Employee
6. employee_reference_rosaton_3
    a. This is a function breaks down the matrix of the numbers from the spread sheet in to employee vectors for the first group of employees in the spread sheet.
        i. Inputs: Employee_Number,e_data
        ii. Outputs: Employee
7. imbedded_rosaton
    a. This is used for the starting function allowing the user to input words as the preferences, makes the program more accessible to people who are not as familiar to MATLAB.
        i. Inputs:
        ii. Outputs: All, One, Introduction
8. n_output_rosaton
    a. This is for the “for” loop in the function; determines the n index for the loop.
        i. Inputs: e_data
        ii. Outputs: n
9. Starting_Function_rosaton
    a. This is the function that the user changes. Essentially, it is for setting preferences in the program so the program calculates what the user wants to see.
        i. Inputs: z
        ii. Outputs: Start, Warnings, Employee_Number

Project Requirement Specs:

Table A

Requirement	Function	Line
Reads a file	exec_function_project_rosaton	19,52
User defined function #1	exec_function_project_rosaton	N/A
User defined function #2	employee_PTO_Bank_rosaton	N/A
User defined function #3	employee_PTO_calculator_rosaton	N/A
User defined function #4	employee_reference_rosato	N/A
User defined function #5	employee_reference_rosaton_2	N/A
User defined function #6	employee_reference_rosaton_3	N/A
User defined function #7	embedded_rosaton	N/A
User defined function #8	n_output_rosaton	N/A
User defined function #9	Starting_Function_rosaton	N/A
For Loop	exec_function_project_rosaton	57
While Loop	employee_PTO_Bank_rosaton	14,15
Embedded Loops	employee_PTO_Bank_rosaton exec_function_project_rosaton	14,15 114-125
Vector/Matrix	employee_PTO_Bank_rosaton exec_function_project_rosaton employee_PTO_calculator_rosaton employee_reference_rosaton employee_reference_rosaton_2 employee_reference_rosaton_3	14-17,28-31 19-32,52-76 7 7-17 8-40 8-58
If/Elseif structure	exec_function_project_rosaton	13

Lines of Code not counting comments: 300

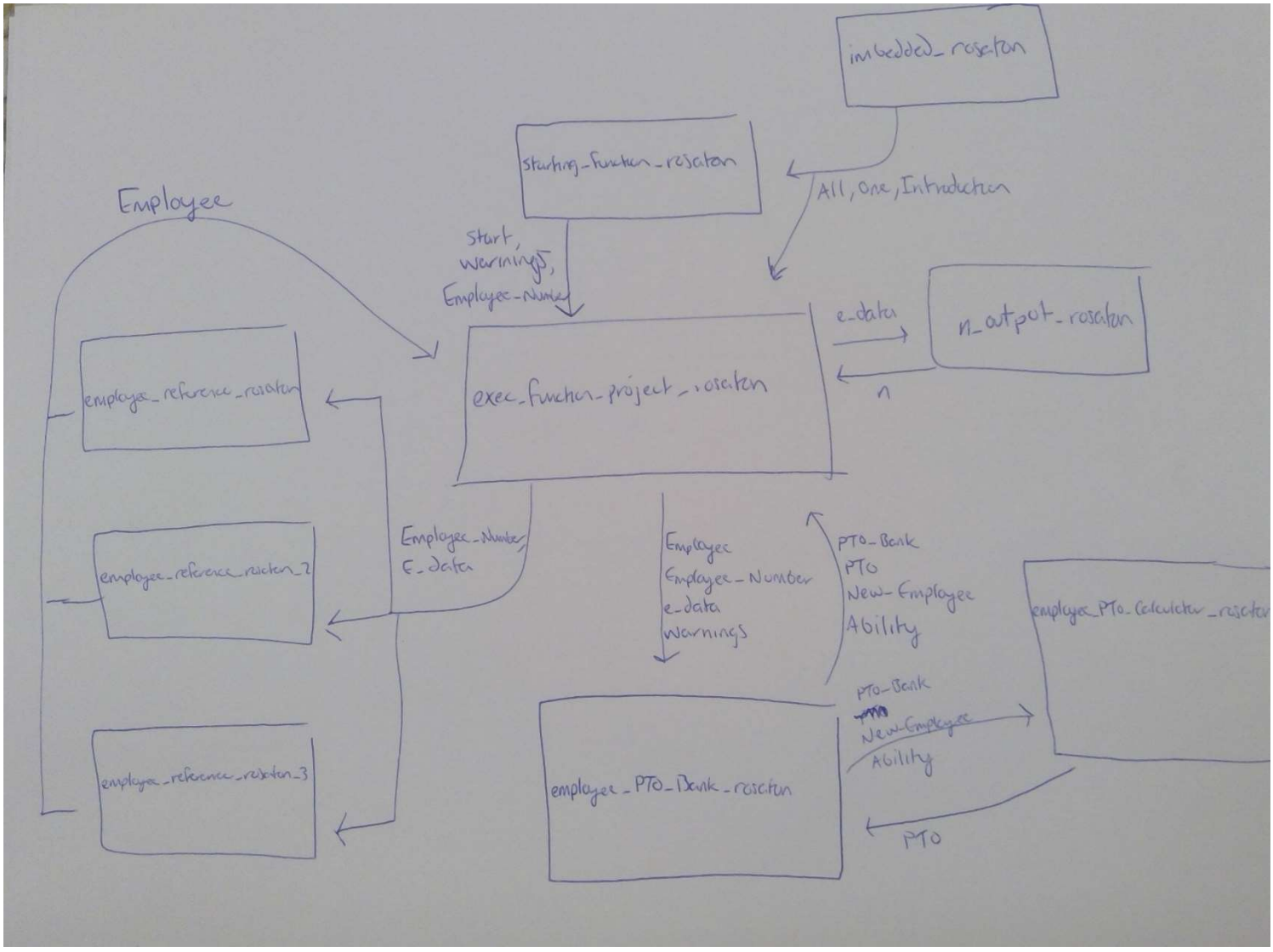
Note: I used a function called "sloc" to count my lines of code in my functions, I got it online. Link to the code share: https://www.mathworks.com/matlabcentral/fileexchange/3900-sloc.

Comments: See code

Note: The lines provided in Table A are not the only instances where the required MATLAB functions are used. Reference the code prints at the bottom of this report if you want to see more instances of the required functions.

Function Diagram:

Diagram B



User Manual:

Welcome to the PTO Calculator! This user guide will provide you with the tools to successfully evaluate paid time off for employees along with the analysis of employee compliance status. The program is designed for the user to set preferences in one function, nothing should be inputted through the command window. Please follow the steps below to successfully be able to run the program.

1. **Initialization of program functions**

   You will want to initialize all 9 of the program's functions to begin. The functions are listed below.

   1. *exec_function_project_rosaton*
   2. *employee_PTO_Bank_rosaton*
   3. *employee_PTO_calculator_rosaton*
   4. *employee_reference_rosaton*
   5. *employee_reference_rosaton_2*
   6. *employee_reference_rosaton_3*
   7. *embedded_rosaton*
   8. *n_output_rosaton*
   9. *Starting_Function_rosaton*

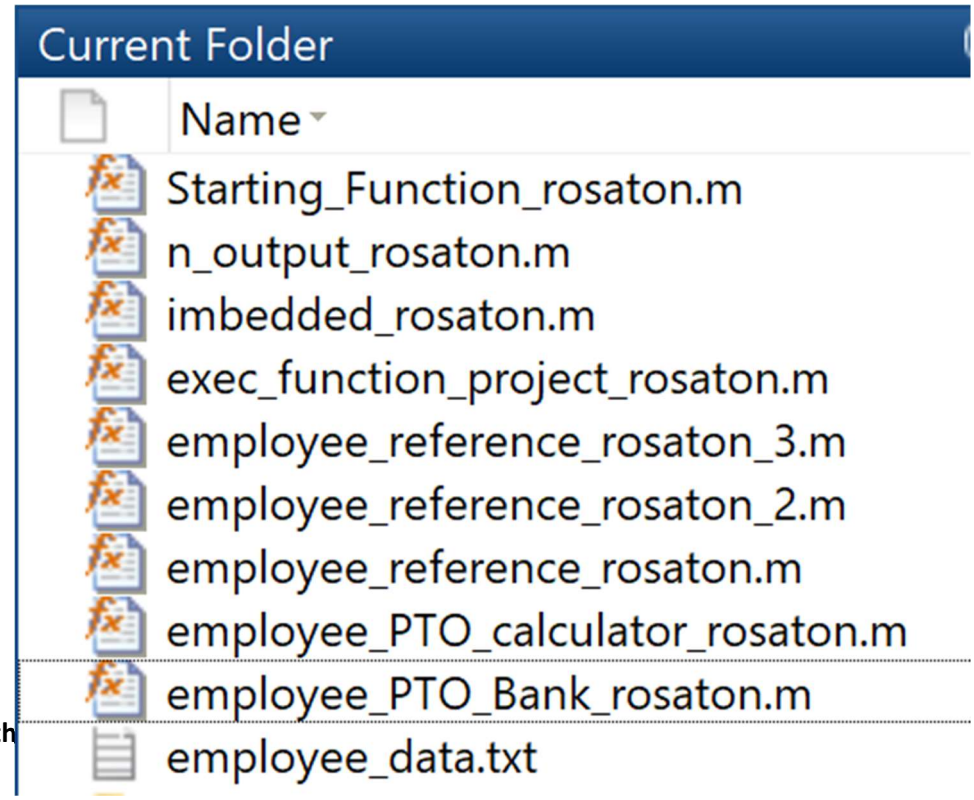
2. **Initialization of the employee data**

   You will want to upload the .txt file provided into the current folder with the functions on MATLAB. The file is called employee_data.txt. This file is designed to change based on the pay cycle. A new file is uploaded every pay cycle.

   Your Current file should look like this:

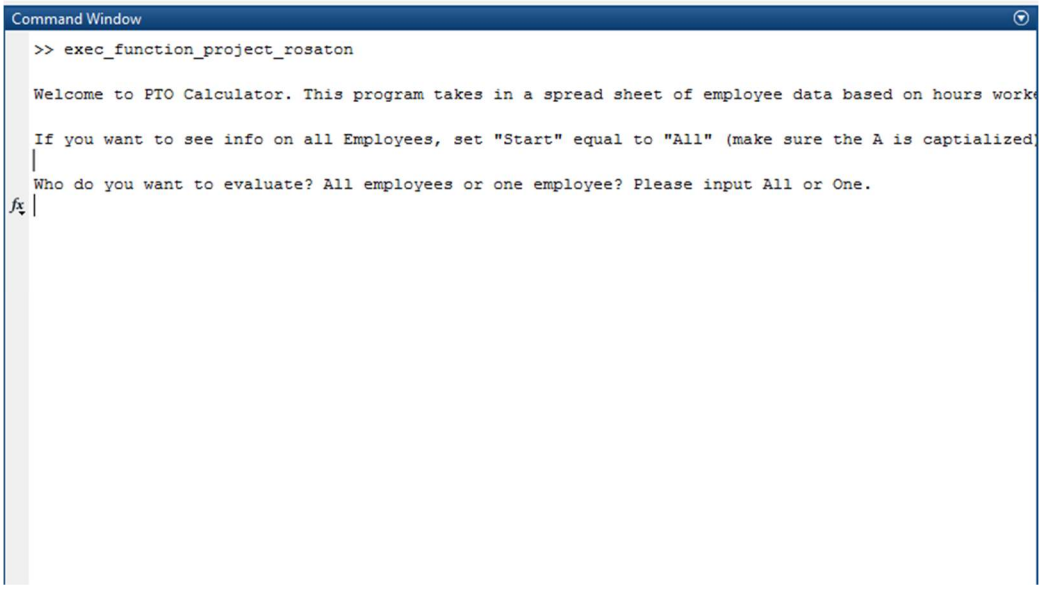
   Picture C

3. **Running th**



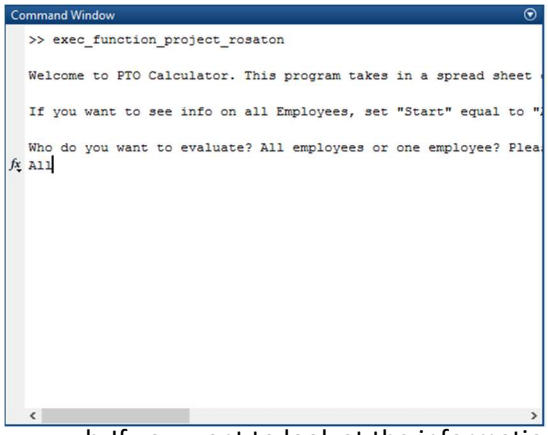
The program will start out in introduction mode. Hit run on *exec_function_project_rosaton* to start the program. The output should be a few print statements. It should look this like below:

Picture D



4. **Now that you have begun the program, you want to set your preferences. You have two options for preferences All or One (employee(s)).**
   a. <u>If you want to look at the data for all employees for PTO earned only</u>, run the executive function and enter "*All"* in the prompt. Make sure the "*A*" in "*All*" is capitalized. The command window will display data on all employees. You will also be able to see data analytics on the set of employees.

Picture E

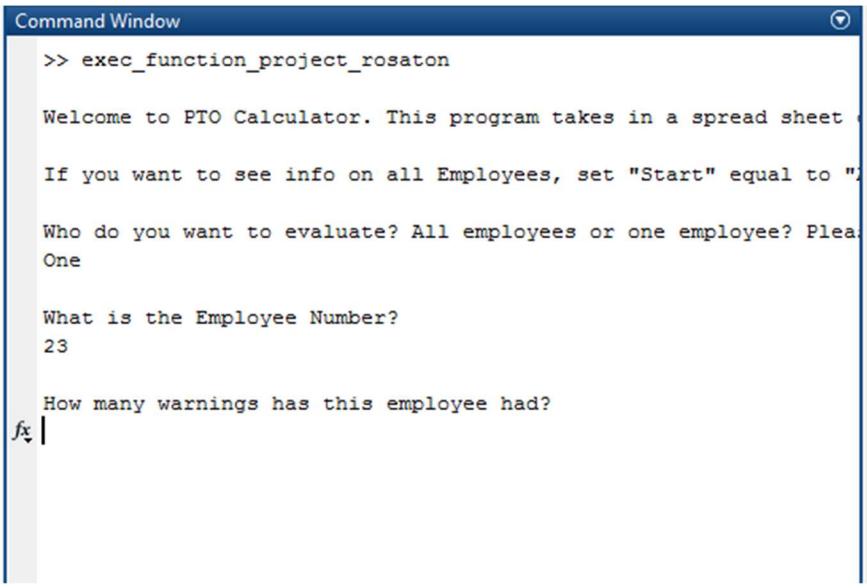


Picture G

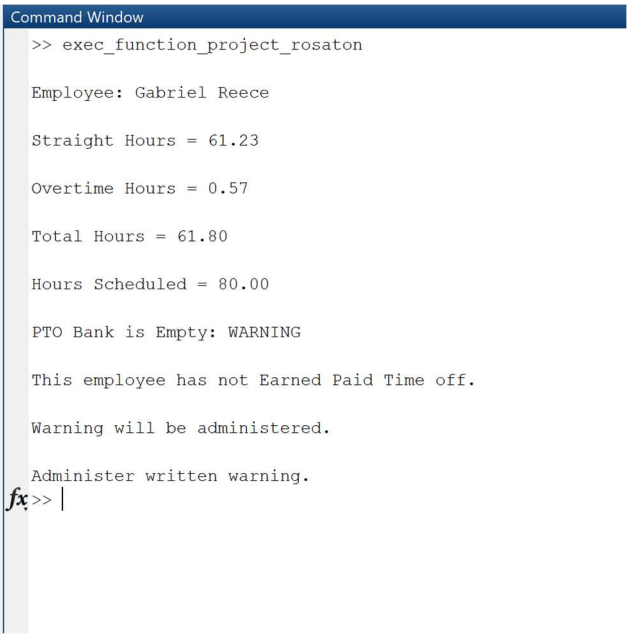
   b. <u>If you want to look at the information and compliance data</u> (what to do based on warnings), enter "One" in the prompt in the

beginning of the program. Now that you have set your preference to *"One"* you will have to input *"Employee_Number"* and "*Warnings"* in the prompt. To pick the employee you want to reference, please refer to the spread sheet provided below. The number of warnings is referencing how many warnings the employee has had for not reaching compliance. Please note that the warnings and earned PTO shown in the spread sheet will be empty as it is for a work place to input. Now that the preferences are set the function should run on its own. The command window will display data on one employee when the start function is set correctly. A sample is below shown in Picture H and Picture I:

Picture H



Picture I



Employee Spreadsheet

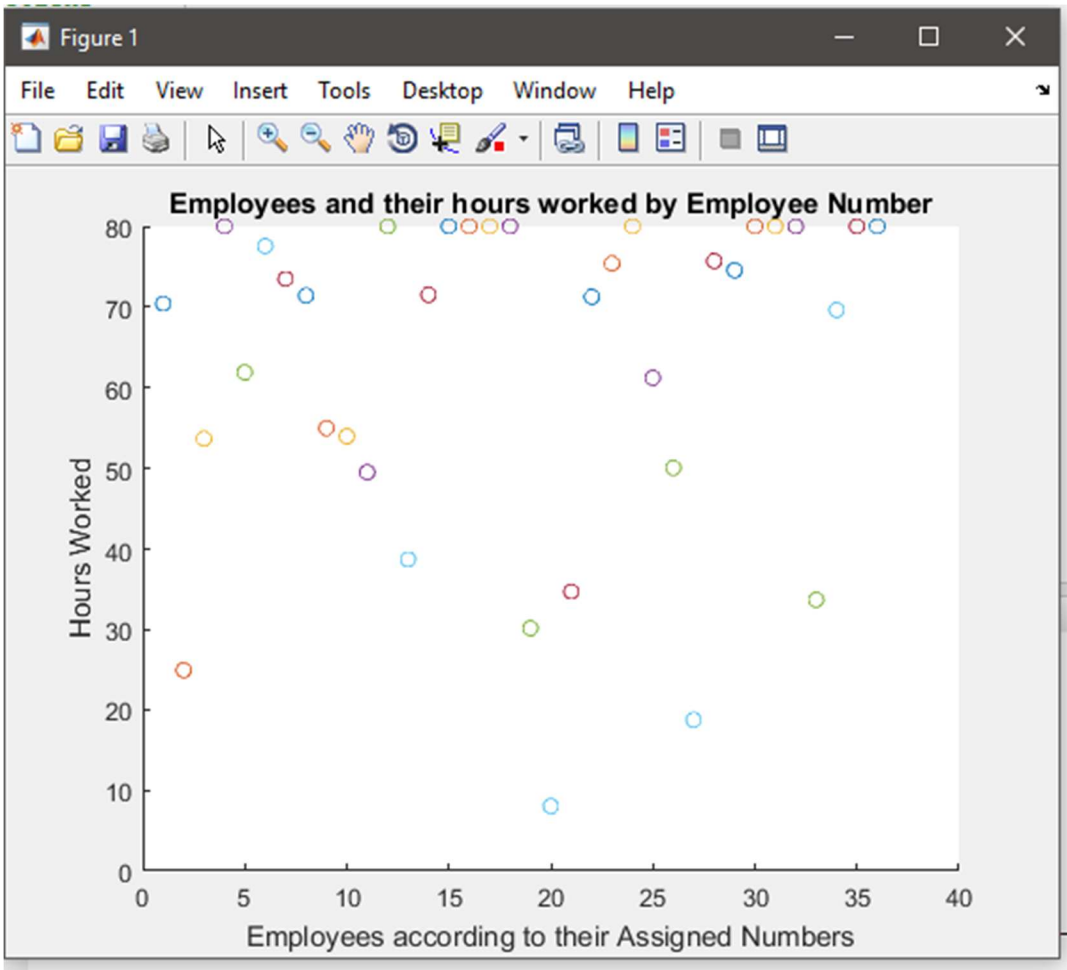
Note: The Employee Number is to the left of the employee name.

Table J

	Employee Reference Sheet	Warnings	Earned PTO?
1	Ball, Ariel		
2	Barnitz, Rachel		
3	Barrett, Shawna		
4	Beard, Taylor		
5	Bhavnani, Daisy		
6	Bing, Elaine		
7	Bogaards, Mackenzie		
8	Bradford, Dorian		
9	Brady, Marge		
10	Cochenour, Alice		
11	Crumpler, Virginia		
12	Daye, Yvonne		
13	Farmwald, Erin		
14	Ferris, Cheyenne		
15	Huda, Bisma		
16	Jones, Teela		
17	Lewis, Mary (KK)		
18	Mazzola, Lydia		
19	Miller, Robert		
20	Mitchley, Amber		
21	Nicholson, Paul		
22	Pace, Elizabeth		
23	Price, Ayesha		
24	Ratnarajah, Yohambigai		
25	Reece, Gabriel		
26	Roberts, Evelyn		
27	Rogers, Alexis		
28	Smith, Corrie		
29	Smith, Kitty		
30	Stallings, Bethany		
31	Sturm, Amy M		
32	Tergerson, Becky		
33	Trasser, Dakota		
34	Williams, Mekye		
35	Willis, Lee		
36	Yeganeh, Farnaz		

Sample of Data Analytics for this Program

Picture K




```
DATA ANALYTICS

A graph to display trends among employee hours will also be displayed.

 The average amount of hours worked is 62.93 hours

 The average amount of overtime worked is 1.38 hours

 The number of employees that hit compliance for this pay cycle is 14 employees
```

Code Print Out:

exec_function_project_rosaton

```
function[] = exec_function_project_rosaton(Start,Warnings,Employee_Number)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132 Program Description
%   This function ...
%   Is the executive function that compiles all of my other functions
%   together to run the code.
% Function Call
%   [] = exec_function_project_rosaton(Start,Warnings,Employee_Number)
%
% Input Arguments
%   1.Start
%   2.Warnings
%   3.Employee_Number
% Output Arguments
%   None
%
% Assignment Information
%   Assignment: Individual Matlab Project
%   Author:  Nicholas Rosato, rosaton
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initialization
%Introduction

prompt = ('\nWould you like to open the user manual? Input Yes or No.
\n');
Yes = 1;
yes = Yes;
No = 2;
Open = input(prompt);
if Open == 1
    winopen Project_Report_rosaton_pdf.pdf
end

z=0;
    fprintf('\nWelcome to PTO Calculator. This program takes in a spread
sheet of employee data based on hours worked and hours scheduled to calculate
paid time off and what to do if the employee is breaking the rules.\n')
    fprintf('\nIf you want to see info on all Employees, set "Start" equal to
"All" (make sure the A is captialized), If you want to look into detail on
one employee, set Start equal to "One" (make sure the O is capitalized).\n')
    [Start,Warnings,Employee_Number] = Starting_Function_rosaton(z);
    [All,One,Introduction] = embedded_rosaton();


%Program In action, there is some initialization throughout the program
%after this line.


    if Start == 1 %If the preference "One" is choosen
    e_data = load('employee_data.txt'); %This is the script the function will
bring in
    if Employee_Number == 0
```

```
        fprintf('\nPlease enter Employee_Number 1-%d in the start
function.\n',length(e_data))
        [Start,Warnings,Employee_Number] = Starting_Function_rosaton()
    elseif Employee_Number < 5
        [Employee] = employee_reference_rosaton(Employee_Number,e_data);
    elseif Employee_Number >=5 & Employee_Number <= 16
        [Employee] = employee_reference_rosaton_2(Employee_Number,e_data);
    elseif (Employee_Number > 16) & (Employee_Number <= length(e_data))
        [Employee] = employee_reference_rosaton_3(Employee_Number,e_data);
    elseif Employee_Number > length(e_data)
            z=1;
            while Employee_Number > length(e_data)
                fprintf('\nPlease enter a valid Employee_Number 1-%d, your
number is too high.\n',length(e_data))
                [Start,Warnings,Employee_Number] =
Starting_Function_rosaton(z);
            end
        if Employee_Number < 5
            [Employee] = employee_reference_rosaton(Employee_Number,e_data);
        elseif Employee_Number >=5 & Employee_Number <= 16
            [Employee] =
employee_reference_rosaton_2(Employee_Number,e_data);
        elseif (Employee_Number > 16) & (Employee_Number <= length(e_data))
            [Employee] =
employee_reference_rosaton_3(Employee_Number,e_data);
        elseif Employee_Number > length(e_data)
        end
    end
    if (Employee_Number > 0) & (Employee_Number <= length(e_data))
    [PTO_Bank,PTO,New_Employee,Ability] =
employee_PTO_Bank_rosaton(Employee,Employee_Number,e_data,Warnings);
    if Ability == 1
        fprintf('\nCongratulations, this employee has earned 2 hours of paid
time off.\n')
    elseif Ability == 0
        fprintf('\nThis employee has not Earned Paid Time off.\n')
        if PTO_Bank < 0
            fprintf('\nWarning will be administered.\n')
            if Warnings == 1
                fprintf('\nAdminister written warning.\n')
            elseif Warnings >= 2
                fprintf('\nConsider termination.\n')
            elseif Warnings == 0
                fprintf('\nAdminister verbal warning.\n')
            end
        end
    end
    end
elseif Start == 2 %If the preference "All is chosen"
    e_data = load('employee_data.txt');
    [n] = n_output_rosaton(e_data);
    Employee_Number = 1
    %This loops through all employees, n is initialized in the n output
    %function
    for i = 1:n;
      fprintf('\nEmployee Number: %d\n', Employee_Number)
       if Employee_Number < 5
```

```
            [Employee] = employee_reference_rosaton(Employee_Number,e_data);
        elseif Employee_Number >=5 & Employee_Number <= 16
            [Employee] =
employee_reference_rosaton_2(Employee_Number,e_data);
        elseif Employee_Number > 16
            [Employee] =
employee_reference_rosaton_3(Employee_Number,e_data);
        elseif Employee_Number > 36
      z = 0;
fprintf('\nPlease enter a valid Employee_Number 1-%d, your number is too
high\n',length(e_data))
            [Start,Warnings,Employee_Number] = Starting_Function_rosaton(z)
        end
        if (Employee_Number > 0) & (Employee_Number < length(e_data))
        [PTO_Bank,PTO,New_Employee,Ability] =
employee_PTO_Bank_rosaton(Employee,Employee_Number,e_data,Warnings);
        if Ability == 1
            fprintf('\nCongratulations, this employee has earned 2 hours of
paid time off.\n')
        elseif Ability == 0
            fprintf('\nThis employee has not earned paid time off.\n')
            if PTO_Bank < 0
                fprintf('\n Warning will be Administered.\n')
                fprintf('\n If you want to see details about warnings with
this employee, set "Start" equal to "One" and enter how many warnings this
employee has had along with the employee''s number in the prompt.\n')
                end
            end
        end
        Employee_Number = Employee_Number + 1
      fprintf('\n----------------------------------------------------------------
----------\n')
    end
fprintf('\nA graph to display trends among employee hours will also be
displayed.\n')
    Employee_Graph = 0;
    for x = 1:length(e_data)
        Employee_Graph = Employee_Graph + 1;
       for y = e_data(Employee_Graph, 1)
           scatter(x,y)
           hold on
           title('Employees and their hours worked by Employee Number')
           xlabel('Employees according to their Assigned Numbers')
           ylabel('Hours Worked')
       end
    end
Average_Hours = mean(e_data(1:36,1));
    fprintf('\n The average amount of hours worked is %0.2f
hours\n',Average_Hours)
Average_Hours = mean(e_data(1:36,1));
    fprintf('\n The average amount of hours worked is %0.2f
hours\n',Average_Hours)
    Average_Overtime = mean(e_data(1:36,2));
    fprintf('\n The average amount of overtime worked is %0.2f hours\n',
Average_Overtime)
   counter = 0;
    for I = 1:36
```

```
        if e_data(I,3) >= e_data(I,4)
            counter = counter + 1;
        end
    end
     fprintf('\n The number of employees that hit compliance for this pay
cycle is %d employees\n', counter)


    end
end
```

Starting_Function_rosaton

```
function[] = exec_function_project_rosaton(Start,Warnings,Employee_Number)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132 Program Description
%   This function ...
%   Is the executive function that compiles all of my other functions
%   together to run the code.
% Function Call
%   [] = exec_function_project_rosaton(Start,Warnings,Employee_Number)
%
% Input Arguments
%   1.Start
%   2.Warnings
%   3.Employee_Number
% Output Arguments
%   None
%
% Assignment Information
%   Assignment: Individual Matlab Project
%   Author:  Nicholas Rosato, rosaton
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initialization
%Introduction
z=0;
    fprintf('\nWelcome to PTO Calculator. This program takes in a spread
sheet of employee data based on hours worked and hours scheduled to calculate
paid time off and what to do if the employee is breaking the rules.\n')
    fprintf('\nIf you want to see info on all Employees, set "Start" equal to
"All" (make sure the A is captialized), If you want to look into detail on
one employee, set Start equal to "One" (make sure the O is capitalized).\n')
    [Start,Warnings,Employee_Number] = Starting_Function_rosaton(z);
    [All,One,Introduction] = imbedded_rosaton();


%Program In action, there is some initialization throughout the program
%after this line.


    if Start == 1 %If the preference "One" is choosen
    e_data = load('employee_data.txt'); %This is the script the function will
bring in
    if Employee_Number == 0
        fprintf('\nPlease enter Employee_Number 1-%d in the start
function.\n',length(e_data))
```

```
        [Start,Warnings,Employee_Number] = Starting_Function_rosaton()
    elseif Employee_Number < 5
        [Employee] = employee_reference_rosaton(Employee_Number,e_data);
    elseif Employee_Number >=5 & Employee_Number <= 16
        [Employee] = employee_reference_rosaton_2(Employee_Number,e_data);
    elseif (Employee_Number > 16) & (Employee_Number <= length(e_data))
        [Employee] = employee_reference_rosaton_3(Employee_Number,e_data);
    elseif Employee_Number > length(e_data)
            z=1;
            while Employee_Number > length(e_data)
                fprintf('\nPlease enter a valid Employee_Number 1-%d, your
number is too high.\n',length(e_data))
                [Start,Warnings,Employee_Number] =
Starting_Function_rosaton(z)
            end
        if Employee_Number < 5
            [Employee] = employee_reference_rosaton(Employee_Number,e_data);
        elseif Employee_Number >=5 & Employee_Number <= 16
            [Employee] =
employee_reference_rosaton_2(Employee_Number,e_data);
        elseif (Employee_Number > 16) & (Employee_Number <= length(e_data))
            [Employee] =
employee_reference_rosaton_3(Employee_Number,e_data);
        elseif Employee_Number > length(e_data)
        end
    end
    if (Employee_Number > 0) & (Employee_Number <= length(e_data))
    [PTO_Bank,PTO,New_Employee,Ability] =
employee_PTO_Bank_rosaton(Employee,Employee_Number,e_data,Warnings);
    if Ability == 1
        fprintf('\nCongratulations, this employee has earned 2 hours of paid
time off.\n')
    elseif Ability == 0
        fprintf('\nThis employee has not Earned Paid Time off.\n')
        if PTO_Bank < 0
            fprintf('\nWarning will be administered.\n')
            if Warnings == 1
                fprintf('\nAdminister written warning.\n')
            elseif Warnings >= 2
                fprintf('\nConsider termination.\n')
            elseif Warnings == 0
                fprintf('\nAdminister verbal warning.\n')
            end
        end
    end
    end
elseif Start == 2 %If the preference "All is chosen"
    e_data = load('employee_data.txt');
    [n] = n_output_rosaton(e_data);
    Employee_Number = 1
    %This loops through all employees, n is initialized in the n output
    %function
    for i = 1:n;
        if Employee_Number < 5
            [Employee] = employee_reference_rosaton(Employee_Number,e_data);
        elseif Employee_Number >=5 & Employee_Number <= 16
```

```
            [Employee] =
employee_reference_rosaton_2(Employee_Number,e_data);
        elseif Employee_Number > 16
            [Employee] =
employee_reference_rosaton_3(Employee_Number,e_data);
        elseif Employee_Number > 36
            fprintf('\nPlease enter a valid Employee_Number 1-%d, your number
is too high\n',length(e_data))
            [Start,Warnings,Employee_Number] = Starting_Function_rosaton(z)
        end
        if (Employee_Number > 0) & (Employee_Number < length(e_data))
        [PTO_Bank,PTO,New_Employee,Ability] =
employee_PTO_Bank_rosaton(Employee,Employee_Number,e_data,Warnings);
        if Ability == 1
            fprintf('\nCongratulations, this employee has earned 2 hours of
paid time off.\n')
        elseif Ability == 0
            fprintf('\nThis employee has not earned paid time off.\n')
            if PTO_Bank < 0
                fprintf('\n Warning will be Administered.\n')
                fprintf('\n If you want to see details about warnings with
this employee, set "Start" equal to "One" and enter how many warnings this
employee has had along with the employee''s number in the prompt.\n')
                end
            end
        end
        Employee_Number = Employee_Number + 1
    end
    end
end
```

employee_PTO_Bank_rosaton

```
function[PTO_Bank,PTO,New_Employee,Ability] =
employee_PTO_Bank_rosaton(Employee,Employee_Number,e_data,Warnings)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132 Program Description
%   This function ...
%   The PTO bank function will calculate the ability of an employee to obtain
%   paid time off for the current pay cycle.
% Function Call
%   [PTO_Bank,PTO,New_Employee,Ability] =
employee_PTO_Bank_rosaton(Employee,Employee_Number,e_data,Warnings)
%
% Input Arguments
%   1.Employee
%   2.Employee_Number
%   3.e_data
%   4.Warnings
% Output Arguments
%   1.PTO_Bank
%   2.PTO
%   3.New_Employee
%   4.Ability
% Assignment Information
%   Assignment: Individual Matlab Project
%   Author:  Nicholas Rosato, rosaton
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Employee_Bank = zeros(length(e_data)); %Gives the option to input hours left
in the employees PTO Bank


%imbedded while loops allows for PTO calculation for each employee
Ability = 1;
n = 1;
Cut = 1;
while n == 1
    while (Employee(1,3) < Employee(1,4)) & (Cut > 0)
        PTO_Bank = Employee_Bank(Employee_Number) - 1;
        Employee(1,3) = Employee(1,3) + 1;
        Ability = 0;
        if PTO_Bank < 0
            fprintf('\nPTO Bank is Empty: WARNING\n')
            Cut = 0;
            PTO = 0;
            New_Employee = 0;
        end
        n = n + 1;
    end
if Cut==1
New_Employee = Employee(1,3) + (Employee(1,4)-Employee(1,3));
Employee(1,3) = New_Employee;

PTO_Bank = Employee_Bank(Employee_Number);

[PTO] = employee_PTO_calculator_rosaton(Employee,PTO_Bank,New_Employee,
Ability, Warnings);

n = n + 1;
end
end
end
```

employee_PTO_calculator_rosaton

```
function[PTO] =
employee_PTO_calculator_rosaton(Employee,PTO_Bank,New_Employee, Ability,
Warnings)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132 Program Description
%   This function ...
%   This function calculates the paid time off of the employee. It also
%   determnes what to administer to the employee if the employee's hours are
%   not meeting expectations of the employer.
% Function Call
%   [PTO] = employee_PTO_calculator_rosaton(Employee,PTO_Bank,New_Employee,
Ability, Warnings)
%
% Input Arguments
%   1.Employee
%   2.PTO_Bank
```

```
%   3.New_Employee
%   4.Ability
%   5.Warnings
% Output Arguments
%   1.PTO
% Assignment Information
%   Assignment: Individual Matlab Project
%   Author:  Nicholas Rosato, rosaton
%%%
```

```
    fprintf('\nEmployee: Shawna Barrett\n')
elseif Employee == e_data(4,:)
    fprintf('\nEmployee: Taylor Beard\n')
end
%for print statements
Straight_Hours = e_data(Employee_Number,1);
Overtime_Hours = e_data(Employee_Number,2);
Total_Hours = e_data(Employee_Number,3);
Hours_Scheduled = e_data(Employee_Number,4);
fprintf('\nStraight Hours = %0.2f\n',Straight_Hours)
fprintf('\nOvertime Hours = %0.2f\n',Overtime_Hours)
fprintf('\nTotal Hours = %0.2f\n', Total_Hours)
fprintf('\nHours Scheduled = %0.2f\n', Hours_Scheduled)




%For the Programmer's Reference



%Ball_Ariel = e_data(1,:);
%Barnitz_Rachel = e_data(2,:);
%Barrett_Shawna = e_data(3,:);
%Beard_Taylor = e_data(4,:);
%Bahvnani_Daisy = e_data(5,:);
%Bing_Elaine = e_data(6,:);
%Bogaards_Mackenzie = e_data(7,:);
%Bradford_Dorian = e_data(8,:);
%Brady_Marge = e_data(9,:);
%Cochenour_Alice = e_data(10,:);
%Crumpler_Virginia = e_data(11,:);
%Daye_Yvonne = e_data(12,:);
%Farmwald_Erin = e_data(13,:);
%Ferris_Cheyenne = e_data(14,:);
%Huda_Bisma = e_data(15,:);
%Jones_Teela = e_data(16,:);
%Lewis_Mary = e_data(17,:);
%Mazzola_Lydia = e_data(18,:);
%Miller_Robert = e_data(19,:);
%Mitchley_Amber = e_data(20,:);
%Nicholson_Paul = e_data(21,:);
%Pace_Elizabeth = e_data(22,:);
%Price_Ayesha = e_data(23,:);
%Ratnarajah_Yohambigai = e_data(24,:);
%Reece_Gabriel = e_data(25,:);
%Roberts_Evelyn = e_data(26,:);
%Rogers_Alexis = e_data(27,:);
%Smith_Corrie = e_data(28,:);
%Smith_Kitty = e_data(29,:);
%Stallings_Bethany = e_data(30,:);
%Sturm_Amy = e_data(31,:);
%Tergerson_Becky = e_data(32,:);
%Trasser_Dakota = e_data(33,:);
%Williams_Mekye = e_data(34,:);
%Willis_Lee = e_data(35,:);
```

```
%Yeganeh_Farnaz = e_data(36,:);
```

employee_reference_rosaton_2

```
function[Employee] = employee_reference_rosaton_2(Employee_Number,e_data)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132 Program Description
%   This function ...
%   Second set of references, this allows function not to mismatch numbers to
%   different employees.
% Function Call
%   [Employee] = employee_reference_rosaton_2(Employee_Number,e_data)
%
% Input Arguments
%   1.Employee_Number
%   2.e_data
% Output Arguments
%   1.Employee
%
% Assignment Information
%   Assignment: Individual Matlab Project
%   Author:  Nicholas Rosato, rosaton
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%indexes matrix brought in by user
Employee = e_data(Employee_Number,:);

if Employee == e_data(5,:)
    fprintf('\nEmployee: Daisy Bahvnani\n')
elseif Employee == e_data(6,:)
    fprintf('\nEmployee: Elaine Bing\n')
elseif Employee == e_data(7,:)
    fprintf('\nEmployee: Mackenzie Bogaards\n')
elseif Employee == e_data(8,:)
    fprintf('\nEmployee: Dorian Bradford\n')
elseif Employee == e_data(9,:)
    fprintf('\nEmployee: Marge Brady\n')
elseif Employee == e_data(10,:)
    fprintf('\nEmployee: Alice Cochenour\n')
elseif Employee == e_data(11,:)
    fprintf('\nEmployee: Virginia Crumpler\n')
elseif Employee == e_data(12,:)
    fprintf('\nEmployee: Yvonne Daye\n')
elseif Employee == e_data(13,:)
    fprintf('\nEmployee: Erin Farmwald\n')
elseif Employee == e_data(14,:)
    fprintf('\nEmployee: Cheyenne Ferris\n')
elseif Employee == e_data(15,:)
    fprintf('\nEmployee: Bisma Huda\n')
elseif Employee == e_data(16,:)
    fprintf('\nEmployee: Teela Jones\n')
end
%for print statements
Straight_Hours = e_data(Employee_Number,1);
Overtime_Hours = e_data(Employee_Number,2);
Total_Hours = e_data(Employee_Number,3);
```

```
Hours_Scheduled = e_data(Employee_Number,4);
fprintf('\nStraight Hours = %0.2f\n',Straight_Hours)
fprintf('\nOvertime Hours = %0.2f\n',Overtime_Hours)
fprintf('\nTotal Hours = %0.2f\n', Total_Hours)
fprintf('\nHours Scheduled = %0.2f\n', Hours_Scheduled)

end
```

employee_reference_rosaton_3

```
function[Employee] = employee_reference_rosaton_3(Employee_Number,e_data)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132 Program Description
%   This function ...
%   Third set of references, this allows function not to mismatch numbers to
%   different employees.
% Function Call
%   [Employee] = employee_reference_rosaton_3(Employee_Number,e_data)
%
% Input Arguments
%   1.Employee_Number
%   2.e_data
% Output Arguments
%   1.Employee
%
% Assignment Information
%   Assignment: Individual Matlab Project
%   Author:  Nicholas Rosato, rosaton
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%indexes matrix inputted by user
if Employee_Number <= length(e_data);
Employee = e_data(Employee_Number,:);

if Employee == e_data(17,:)
    fprintf('\nEmployee: Mary Lewis\n')
elseif Employee == e_data(18,:)
    fprintf('\nEmployee: Lydia Mazzola\n')
elseif Employee == e_data(19,:)
    fprintf('\nEmployee: Robert Miller\n')
elseif Employee == e_data(20,:)
    fprintf('\nEmployee: Amber Mitchley\n')
elseif Employee == e_data(21,:)
    fprintf('\nEmployee: Paul Nicholson\n')
elseif Employee == e_data(22,:)
    fprintf('\nEmployee: Elizabeth Pace\n')
elseif Employee == e_data(23,:)
    fprintf('\nEmployee: Ayesha Price\n')
elseif Employee == e_data(24,:)
    fprintf('\nEmployee: Yohambigai Ratnarajah\n')
elseif Employee == e_data(25,:)
    fprintf('\nEmployee: Gabriel Reece\n')
elseif Employee == e_data(26,:)
    fprintf('\nEmployee: Evelyn Roberts\n')
elseif Employee == e_data(27,:)
    fprintf('\nEmployee: Alexis Rogers\n')
elseif Employee == e_data(28,:)
    fprintf('\nEmployee: Corrie Smith\n')
```

```
elseif Employee == e_data(29,:)
    fprintf('\nEmployee: Kitty Smith\n')
elseif Employee == e_data(30,:)
    fprintf('\nEmployee: Bethany Stallings\n')
elseif Employee == e_data(31,:)
    fprintf('\nEmployee: Becky Tergerson\n')
elseif Employee == e_data(32,:)
    fprintf('\nEmployee: Amy Sturm\n')
elseif Employee == e_data(33,:)
    fprintf('\nEmployee: Dakota Trasser\n')
elseif Employee == e_data(34,:)
    fprintf('\nEmployee: Mekye Williams\n')
elseif Employee == e_data(35,:)
    fprintf('\nEmployee: Lee Willis\n')
elseif Employee == e_data(36,:)
    fprintf('\nEmployee: Farnaz Yeganeh\n')
end

%for print statements
Straight_Hours = e_data(Employee_Number,1);
Overtime_Hours = e_data(Employee_Number,2);
Total_Hours = e_data(Employee_Number,3);
Hours_Scheduled = e_data(Employee_Number,4);
fprintf('\nStraight Hours = %0.2f\n',Straight_Hours)
fprintf('\nOvertime Hours = %0.2f\n',Overtime_Hours)
fprintf('\nTotal Hours = %0.2f\n', Total_Hours)
fprintf('\nHours Scheduled = %0.2f\n', Hours_Scheduled)

else Employee_Number > length(e_data);
    fprintf('\nPlease input a valid employee number 1-%d\n',length(e_data))
end

end
```

embedded_rosaton

```
function[All,One,Introduction] = embedded_rosaton()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132 Program Description
%   This function ...
%   %This is used for the starting function allowing the user to input
%   words as the preferences, makes the program more accessible to people who
%   are not as familiar to matlab.
% Function Call
%   [All,One,Introduction] = imbedded_rosaton()
%
% Input Arguments
%   None
% Output Arguments
%   1.All
%   2.One
%   3.Introduction
% Assignment Information
%   Assignment: Individual Matlab Project
%   Author:  Nicholas Rosato, rosaton
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Look at exec function if statements to understand this reference better
```

```
All = 2;
Introduction = 0;
One = 1;
end
```

n_output_rosaton

```
function[n] = n_output_rosaton(e_data)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132 Program Description
%   This function ...
%   This is used for the for loop if 'all' is chosen as a preference.
%   It is for the calculation of n.
% Function Call
%   [n] = n_output_rosaton(e_data)
%
% Input Arguments
%   1.e_data
% Output Arguments
%   1.n
%
% Assignment Information
%   Assignment: Individual Matlab Project
%   Author:  Nicholas Rosato, rosaton
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = length(e_data);  %uses the number of rows in the matrix
end
```