**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
   1. Compiler function in this program; connects all functions together. This is the only function that should be ran in the code.
      1. Inputs: z
      2. Outputs: Start, Warnings, Employee\_Number
2. employee\_PTO\_Bank\_rosaton
   1. 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.
      1. Inputs: Employee, Employee\_Number, e\_data, Warnings
      2. Outputs: PTO\_Bank, PTO, New\_Employee, Ability
3. employee\_PTO\_calculator\_rosaton
   1. This function calculates if a specific employee gets paid time off by meeting a set of specific criteria from Northwest Christian Childcare.
      1. Inputs: Employee, PTO\_Bank, New\_Employee, Ability, Warnings
      2. Ouputs: PTO
4. employee\_reference\_rosaton
   1. 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.
      1. Inputs: Employee\_Number, e\_data
      2. Outputs: Employee
5. employee\_reference\_rosaton\_2
   1. 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.
      1. Inputs: Employee\_Number, e\_data
      2. Outputs: Employee
6. employee\_reference\_rosaton\_3
   1. 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.
      1. Inputs: Employee\_Number,e\_data
      2. Outputs: Employee
7. embedded\_rosaton
   1. 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.
      1. Inputs:
      2. Outputs: All, One, Introduction
8. n\_output\_rosaton
   1. This is for the “for” loop in the function; determines the n index for the loop.
      1. Inputs: e\_data
      2. Outputs: n
9. Starting\_Function\_rosaton
   1. 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.
      1. Inputs: z
      2. 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 |

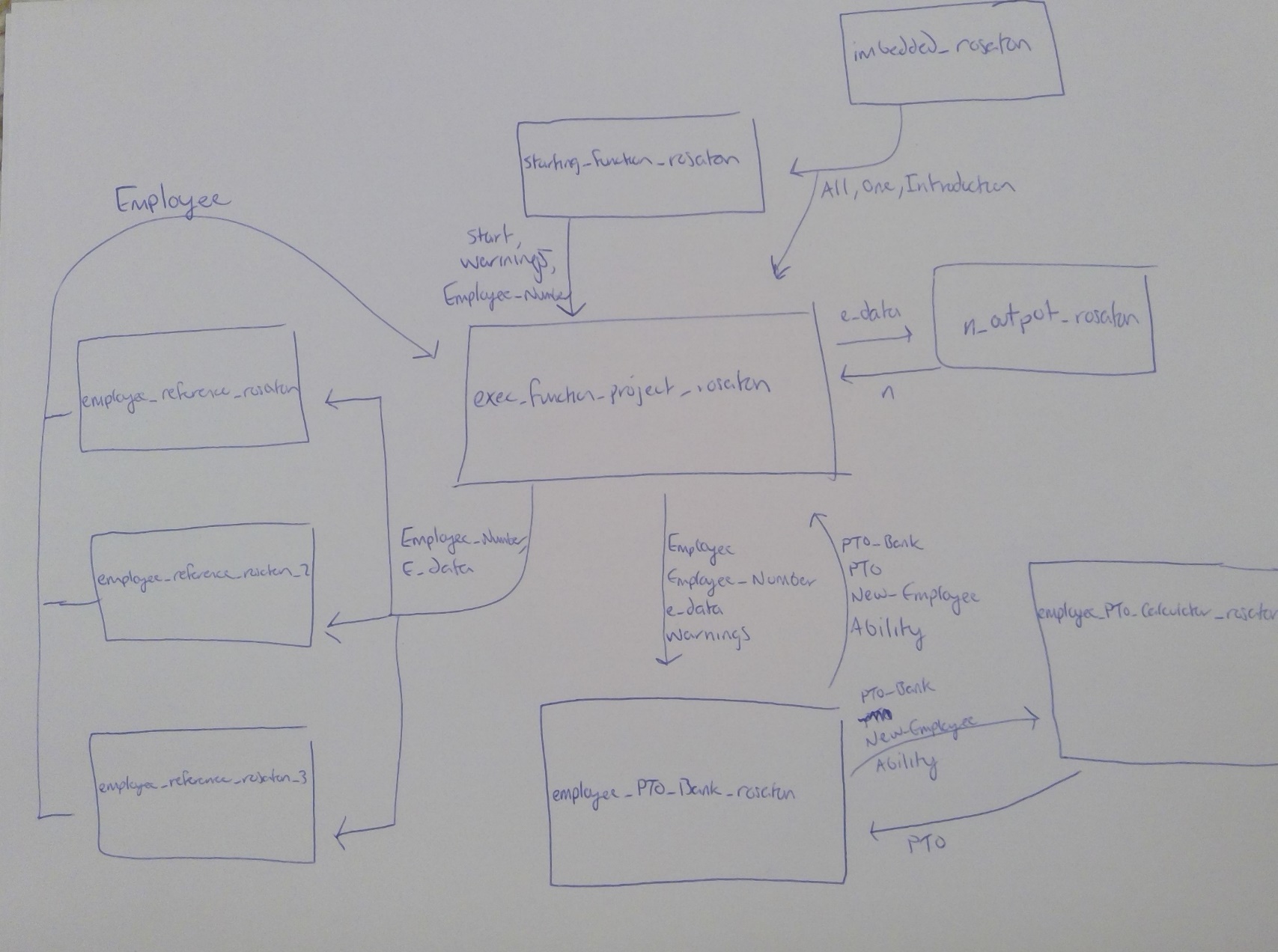
Lines of Code not counting comments: 300

*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**

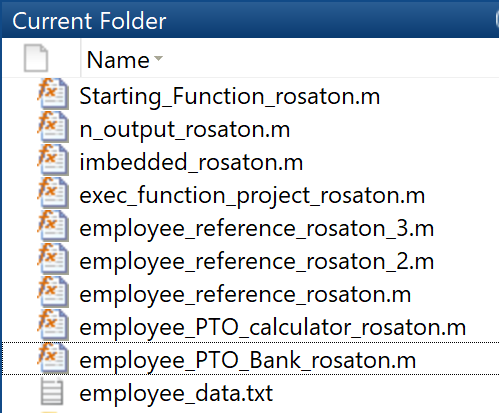
You will want to initialize all 9 of the program’s functions to begin. The functions are listed below.

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*
10. **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**



1. **Running the Program**

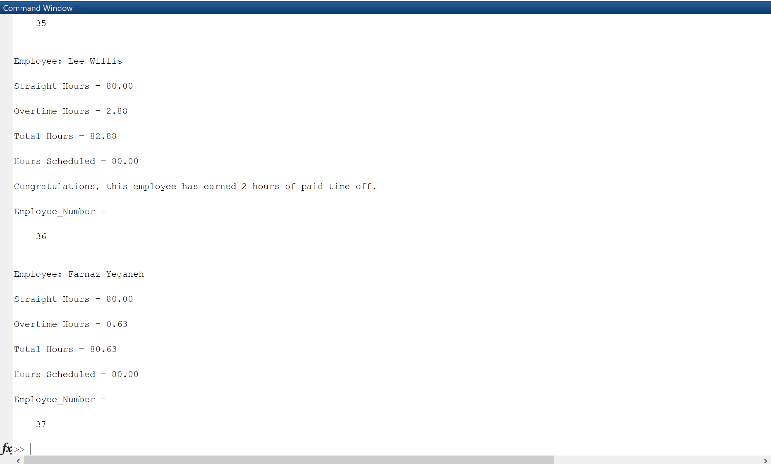
The program will start out in introduction mode. Hit run on *exec\_function\_project\_rosaton* to start the program. The output should be a few print statements. It should look this like below:

A screenshot of a social media post

Description generated with very high confidence**Picture D**

1. **Now that you have begun the program, you want to set your preferences. You have two options for preferences All or One (employee(s)).** 
   1. If you want to look at the data for all employees for PTO earned only, 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**



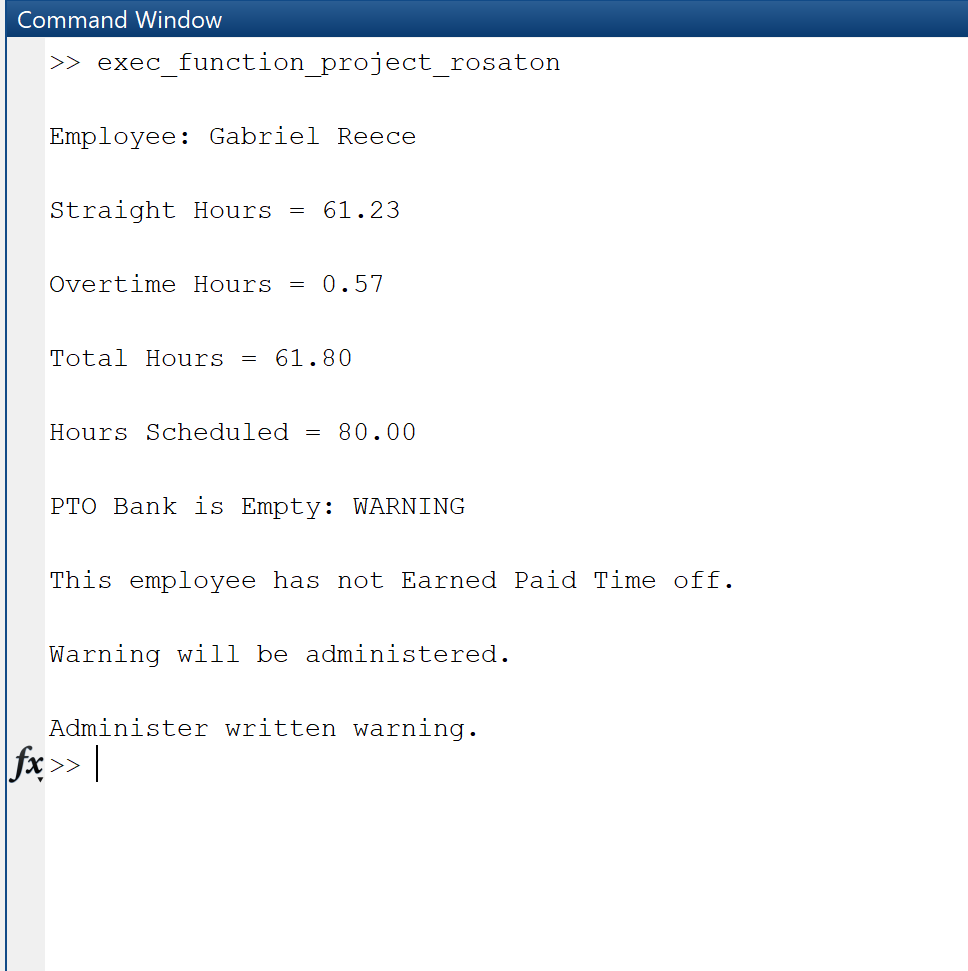
A screenshot of a social media post

Description generated with very high confidence

* 1. If you want to look at the information and compliance data (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:

A screenshot of a social media post

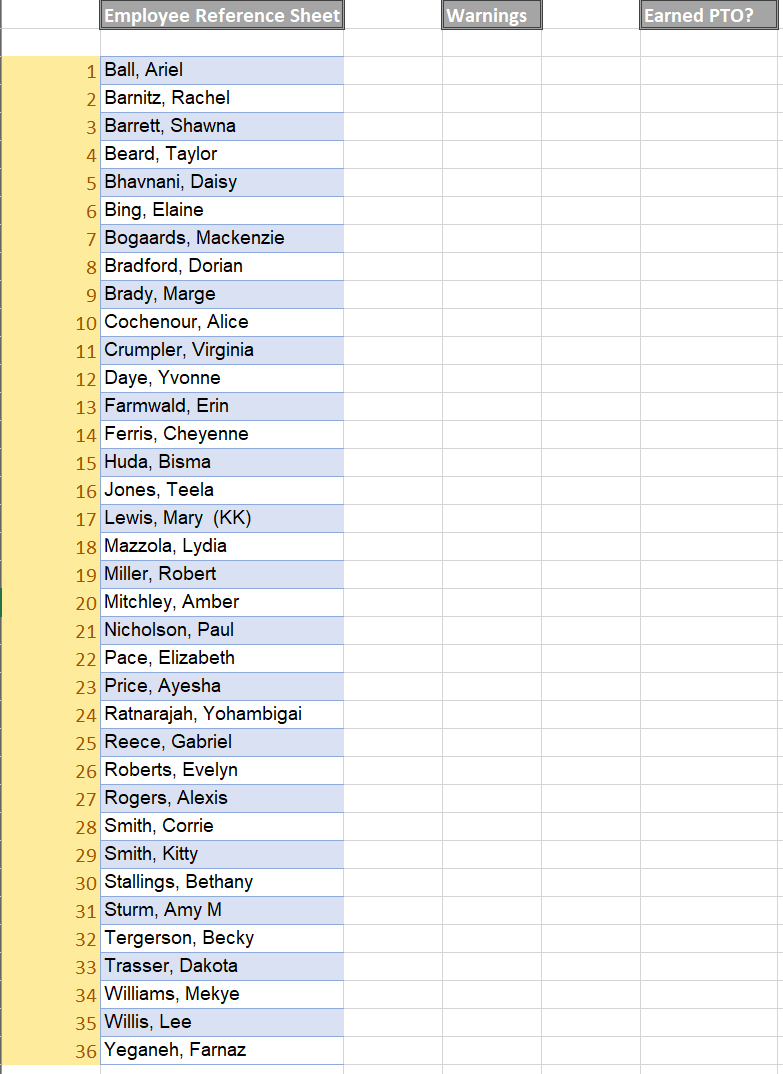
Description generated with very high confidence**Picture H Picture I**



**Employee Spreadsheet**

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

**Table J**



**Sample of Data Analytics for this Program**

**Picture K**

**A screenshot of a social media post

Description generated with very high confidence**

**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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%each employee has a a PTO bank

if (Employee(1,3) >= Employee(1,4)) | (New\_Employee >= Employee(1,4))

PTO = 2 + PTO\_Bank;

if Ability == 0

PTO = PTO\_Bank - 2;

if Warnings == 0

fprintf('\nAdminister Verbal Warning\n')

elseif Warnings == 1

fprintf('\nAdminister Written Warning\n')

elseif Warnings == 2

fprintf('\nConsider Termination of Employee\n')

end

end

end

**employee\_reference\_rosaton**

function[Employee] = employee\_reference\_rosaton(Employee\_Number,e\_data)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ENGR 132 Program Description

% This function ...

% First set of references, this allows function not to mismatch numbers to

% different employees.

% Function Call

% [Employee] = employee\_reference\_rosaton(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(1,:)

fprintf('\nEmployee: Ariel Ball\n')

elseif Employee == e\_data(2,:)

fprintf('\nEmployee: Rachel Barnitz\n')

elseif Employee == e\_data(3,:)

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