Table of Contents

```
function PS06_salt_exec_008_14(cone_salt_weight, piles_salt_weight)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% Calculates the cone height, cone weight, windrow height, and windrow
% weight based off the total amount of salt weight and windrow weight.
%
% Function Call
% function [cone_height, cone_weight, windrow_height, windrow_weight]
 = PS06_salt_exec_008_14(salt_weight, windrow_weight)
%
% Input Arguments
%
% Output Arguments
%
% Assignment Information
%   Assignment:   PS 06, Problem 2
%   Team ID:   008-14
%   Team Member:  Matthew Wen, wen101@purdue.edu
%   Team Member:  Dong Lee, lee3034@purdue.edu
%   Team Member:  John Chapla, jchapla@purdue.edu
%   Team Member:  Ranjan Behl, rbehl@purdue.edu
%   Contributor:  Name, login@purdue [repeat for each]
%   Our contributor(s) helped us:
%     [ ] understand the assignment expectations without
%         telling us how they will approach it.
%     [ ] understand different ways to think about a solution
%         without helping us plan our solution.
%     [ ] think through the meaning of a specific error or
%         bug present in our code without looking at our code.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INITIALIZATION

```
total_salt = 24361; % the total amount of salt for cone and windrow
cone_diameter = 25; % the cone's diamater(m)
windrow_width = 20; % the windrow's width(m)
windrow_length = 48; % the windrow's length(m)
```

CALCULATIONS & FORMATTED TEXT

```
[cone_height, cone_weight] =
 PS06_salt_cone_rbehl_jchapla(cone_diameter); % getting the cone
 height (m) and cone weight (metric ton) from user defined function
[windrow_height, windrow_weight] =
 PS06_salt_cone_wen101_lee3034(windrow_width, windrow_length); %
 getting the windrow height (m) and windrow weight (metric ton) from
 user defined function

num_cone = round((total_salt / cone_weight) + 0.5, 0); % getting the
 number of cones used to carry a specific amount of weight of salt
num_windrow = round((total_salt / windrow_weight) + 0.5, 0); % getting
 the number of windrow used to carry a specific amount of weight of
 salt

fprintf("The user needs %d Cone Piles\n", num_cone); % printing the
 amount of cones needed based off amount of salt given for cones.
fprintf("The user needs %d Windrow Piles\n", num_windrow); % printing
 the amount of windrows needed based off amount of salt given for
 window.
```

```
The height of the concial pile is 7.81 m and its weight is 1670.2 mt.
The height of one windrow pile is 6.25 m and its weight is 3919.6 mt.
The user needs 15 Cone Piles
The user needs 7 Windrow Piles
```

COMMAND WINDOW OUTPUT

The height of the concial pile is 7.81 m and its weight is 1670.2 mt. The height of one windrow pile is 6.25 m and its weight is 3919.6 mt. The user needs 15 Cone Piles The user needs 7 Windrow Piles

ANALYSIS

-- Q1

The PS01 assignment asked for the height for one conical pile, height of one windrow pile, and the weight fo 5 conical piles and the weight fro 2 windrow piles. This assigment prented ount the height of conical piles and weight of conical piles, the height of one windrow piles and the weight of one windrow piles, and the number of conical piles and windrow piles needed rounded up. The most noticable difference that that the command window did not save all the initalized values when it is executed. Once the function ends, all the variables created is removed from memory.

-- Q2

When we call PS06_salt_cone_rbehl_jchapla(21.5), we get the first print statment that we get when we print out the exective statement. We also get the value for cone weight and cone height. When we run the executive statement, we save the value for cone height and cone weight. If we run the exective function, the cone height and cone weight is not stored in a local variable.

-- Q3

We see the header for that function. This is really useful because if the coder does not know the purpose or how to use a function, call help will give the function description, what paramters to put in, and how to use it.

ACADEMIC INTEGRITY STATEMENT

We have not used source code obtained from any other unauthorized source, either modified or unmodified. Neither have we provided access to our code to another. The function we are submitting is our own original work.

Published with MATLAB® R2018b