Enggen131 Good Practice Guidelines

Acceptable

You used some of the project specification text in the header comments for your function

```
% wholeNumberAverage function is responsible for checking whether or not
% the average of the digits is a whole number
% Inputs:   s, an array of numbers representing a sequence to check
% Outputs:  b, zero if the array does NOT average to a whole number,
%           otherwise it returns the whole number average (which corresponds
%           to the number of balls required to juggle the pattern)
function b = wholeNumberAverage(s)

% calculate average
av = mean(s);


% check if the rounded average is equal to the original average
roundedAv = round(av);
if av > 0 && roundedAv == av
    b = av;
else
    b = 0;
end
```

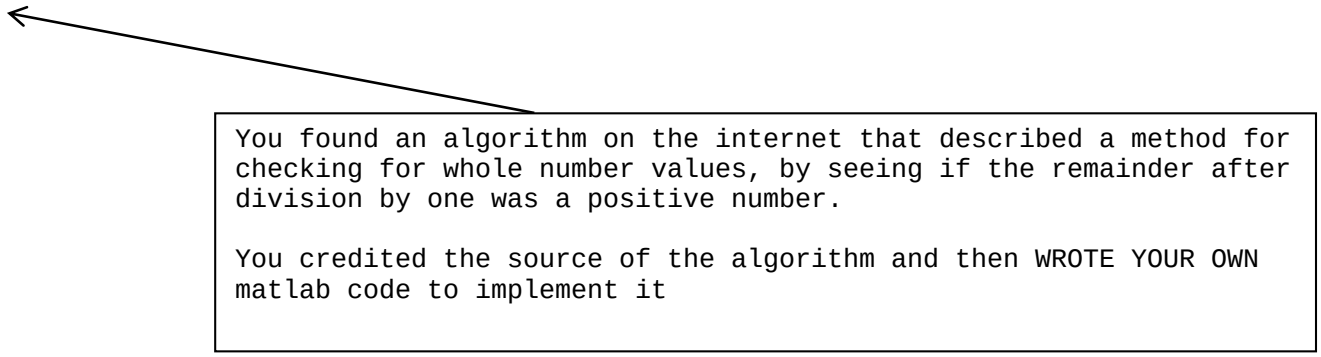
A friend told you about the mean function, which can be used to find the average of an array. You looked up the matlab help on mean and decide to use it, then WROTE YOUR OWN CODE

A group of you discussed ideas about how to check if a value is a whole number. As part of the group discussion, everyone figured out that if the original value is the same as the rounded value, then it must be a whole number. You then WROTE YOUR OWN CODE, using this idea.

Acceptable

```
% wholeNumberAverage function is responsible for checking whether or not
% the average of the digits is a whole number
% Inputs:   s, an array of numbers representing a sequence to check
% Outputs:  b, zero if the array does NOT average to a whole number,
%           otherwise it returns the whole number average (which corresponds
%           to the number of balls required to juggle the pattern)
function b = wholeNumberAverage(s)


% check if the rounded average is equal to the original average
% by seeing if the reminder after division by one is positive
% algorithm retrieved from www.mycoolalgorithm.com/aGreatIdea
if rem(s,1) > 0
    b = mean(s);
else
    b = 0;
end
```



Unacceptable – DO NOT DO

```
% checks whether a string consists of nonzero digits only
% (i.e. the characters 1 to 9 inclusive)
% Inputs:   s, a string to check
% Outputs:  isValid, a Boolean variable, true if the string contains only
%           non zero digits and false otherwise
function isValid = nonZeroDigitsOnly(s)

isValid = true;
i = 1;

% loop through characters to see if they are within a valid ascii range
while isValid==true && i <= length(s)
    if (s(i) < 49 || s(i) > 57)
        % if out of range, this sequence is invalid
        isValid = false;
    end
    i = i +1;
end
```

You were unsure what to write for the function header comments so took a photo of your neighbour's work with your iphone and then typed in their comment lines later

You were having trouble late at night with debugging your while loop. You asked your friend about it on msn and they sent through a few lines of their working code, so you used their while loop condition.

You didn't know how to write an if statement that worked, so your friend came to your computer and typed out this line for you

Unacceptable – DO NOT DO

```
% takes a string containing digits and coverts it to the corresponding
% array of characters
% Inputs:       c, a string of characters containing only nonzero digits
% Outputs:      n, an array of numbers corresponding to the digits of the string
function [n] = char2num(c)

% convert to an array of numbers
for i=1:length(c)
    n(i) = str2num(c(i));
end
```

You didn't know how to write this function, so a friend emailed you their code, which you then copied. To make it look different you changed the wording of the comments and the variable names

```
% generate carry path starting at height 0, going from an initial x position
% to a final x position (specified in terms of cm from origin)
% While this could be a straight line, it looks prettier if the ball dips
% down below the y axis
%
% Inputs:   1) initial x co-ordinate (catch location)
%           2) final x co-ordinate (throw location)
% Outputs:  1) An array of x co-ordinates for the path through the hand
%           2) An array of y co-ordinates for the path through the hand
function [x,y]= generateParabolicPath(xinit,xfinal)

x = linspace(xinit,xfinal,5);
y = [0 -1 -2 -1 0];
```

You couldn't write this function by yourself so you and a friend worked on it together. After writing the entire thing together you each took a copy and put it into your project.