

Henry Meyerson  
109190761  
Assignment 2  
Zagrodzki; CSCI 1320-112

## Assignment 2 – Calculator +

### Task 1) Pseudocode

Divide first variable by second variable, store as variable z

If first variable is less than variable z, AND second variable is greater than variable y

Return True

Else

Return False

% As an aside, I don't believe it is ever possible for this function to return TRUE

MeyersonHenryCalcScriptA.m

```
%Author: Henry Meyerson
%Date: Sept 10, 2018

clear %clear the workspace
clc %clear the command window
close all % close all windows (graphs)

%INPUTS
% Input A
varA = input('Enter First Number: ');
% Input B
varB = input('Enter Second Number: ');

%OUTPUTS
%Addition
addition = varA + varB;
fprintf('%.1f + %.1f = %.1f \n',varA,varB, addition);

%Subtraction
subtraction = varA - varB;
fprintf('%.1f - %.1f = %.1f \n',varA,varB, subtraction);

%Multiplication
multiplication = varA * varB;
fprintf('%.1f * %.1f = %.1f \n',varA,varB, multiplication);

%Division
division = varA / varB;
fprintf('%.1f / %.1f = %.1f \n',varA,varB, division);

%Power
power = varA ^ varB;
fprintf('%.1f ^ %.1f = %.1f \n',varA,varB, power);

%Task 1
T1 = varA < division && varB > division;
fprintf('Task 1) %d \n',T1);
```

## Task 2) Functions

addFunc.m

```
function sum = addFunc(a, b)
sum = a + b;
end
```

subtractFunc.m

```
function difference = subtractFunc(a, b)
difference = a - b;
end
```

multFunc.m

```
function product = multFunc(a, b)
product = a * b;
end
```

divFunc.m

```
function quotient = divFunc(a, b)
quotient = a / b;
end
```

powerFunc.m

```
function expo = powerFunc(a, b)
expo = a ^ b;
end
```

quotientInBoundsFunc.m

```
function parity = quotientInBoundsFunc(a, b)
c = a / b;
parity = a < c && b > c;
end
```

### Task 3) Script

meyersonHenryCalcScriptB.m

```
%Assignment 2 - Calculator +
%Version: B
%Author: Henry Meyerson
%Date: Sept 13, 2018

clear %clear the workspace
clc %clear the command window
close all % close all windows (graphs)

%INPUTS
% Input A
varA = input('Enter First Number: ');
% Input B
varB = input('Enter Second Number: ');

%OUTPUTS
%Addition
addition = addFunc(varA,varB);
fprintf('%.1f + %.1f = %.1f \n',varA,varB, addition);

%Subtraction
subtraction = subtractFunc(varA,varB);
fprintf('%.1f - %.1f = %.1f \n',varA,varB, subtraction);

%Multiplication
multiplication = multFunc(varA, varB);
fprintf('%.1f * %.1f = %.1f \n',varA,varB, multiplication);

%Division
division = divFunc(varA,varB);
fprintf('%.1f / %.1f = %.1f \n',varA,varB, division);

%Power
power = powerFunc(varA,varB);
fprintf('%.1f ^ %.1f = %.1f \n',varA,varB, power);

%Task 1
T1 = quotientInBoundsFunc(varA,varB);
fprintf('Task 1) %d \n',T1);
```

## Task 4) Scripts vs Functions – Carnot Efficiency

carnot.m

```
%Assignment 2 - Carnot Efficiency Calculator
%Version: A
%Author: Henry Meyerson
%Date: Sept 14, 2018

clear %clear the workspace
clc %clear the command window
close all % close all windows (graphs)

%INPUTS
% Input TC
TC = input('TC: ');
% Input TH
TH = input('TH: ');

%Run Calc Function
n = CarnotE(TC,TH);

%Print Output
fprintf('Maximum Efficiency: %.3f\n',n)
```

CarnotE.m

```
function n = CarnotE(TC,TH)
n = 1-(TC/TH);
end
```

### Task 5) Logical Statements

A) There are 8 possible combinations ( $2^3$ ), of the 3 logical variables

B)  $p = \text{xor}(X \ \&\& \ Y, \sim Z)$

or

$p = (X \ \&\& \ Y) \ || \ (\sim Z)$

The instructions are unclear as to whether the or is an “exclusive or” or not

C)

X	Y	Z	$(X \ \&\& \ Y) \    \ (\sim Z)$
T	T	T	T
T	T	F	<u>T</u>
T	F	T	F
T	F	F	T
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	T

X	Y	Z	$\text{xor}(X \ \&\& \ Y, \sim Z)$
T	T	T	T
T	T	F	<u>F</u>
T	F	T	F
T	F	F	T
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	T

Underlined is the only difference between these two tables depending on the exclusive or.