Assignment 7

Juan Silva

ECE 309

Mar. 28, 2018

***Problem Set 11***

**Problem Set 11-1:**

Modify the finance code from p.7 to create a function M-file *invest* that

•takes inputs: princ (the # of dollars to be initially deposited in a bank account), princ\_inc (the # of dollars you will be adding to the account at the end of each year), princ\_des(the # of dollars you want to have at the end of the investment period), and perc(the annual percentage of interest paid by the bank).

•outputs n\_yrs(the # of years it will take your investment to reach princ\_desdollars).

Check your code by finding the # of years needed to accumulate at least $10,000 if we initially deposit $500 and add another $500 at the end of each year, and the bank pays 5% annually. (Ans: 14 yrs)

**Main Script**

% Program Name: finance\_TEST.m

% Author: Juan Silva Last Modified: Mar. 27, 2018

% Description: This program prompts the user to enter investment value information and calculates the number of years necessary to reach the final investment period.

clear, clc, close all

format short, format long

princ = input('Enter principle: ');

princ\_inc = input('Enter annual principle: ');

princ\_des = input('Enter projection accumulated: ');

perc = input('Enter interest percent: ');

n\_yrs = finance(princ,princ\_inc,princ\_des,perc);

fprintf('It will take %d years to accumulate $%d \n', n\_yrs, princ\_des);

**Function Script**

% Program Name: finance.m

% Author: Juan Silva Last Modified: Mar. 28, 2018

% Description: This function will calculate the number of years to reach the investment period based on the inputs.

function n\_yrs = finance(princ,princ\_inc,princ\_des,perc)

tax = perc / 100; %percent conversion

n\_yrs = 0; %years

while princ < princ\_des

n\_yrs = n\_yrs + 1;

princ = princ + (princ \* tax) + princ\_inc;

end

end

**Results**

Enter principle: 500

Enter annual principle: 500

Enter projection accumulated: 10000

Enter interest percent: 5

It will take 14 years to accumulate $10000

**Problem Set 11-2:**

Modify your code triangles (from previous assignment) to allow a user to input the triangle sides a, b, and c.

Further, modify your code triangles (from previous problem 2) to check that the user input values for a, b, and c meet the constraint that holds for all triangles:  
 The SUM of any TWO SIDES MUST BE GREATER THAN THE LENGTH OF THE THIRD SIDE. In other words, (with sides a, b, and c), all 3 of the following must be true:a + b > c, b + c > a, and a + c > b  
 Your code should only execute if the values for a, b, and c meet the above constraint; otherwise, it should put out an error statement, and halt execution.

**Main Script**

% Program Name: triangle\_TEST.m

% Author: Juan Silva Last Modified: Mar. 26, 2018

% Description: This program prompts the user to enter three values for the sides of a triangle. The program will output the area and perimeter of a triangle.

clear, clc

format short, format compact

a = input('Enter value for side a: ');

b = input('Enter value for side b: ');

c = input('Enter value for side c: ');

[per,area] = triangles(a,b,c);

fprintf('The perimeter of a triangle with sides %d, %d, %d equals %d.\n', a, b, c, per)

fprintf('The area of a triangle with sides %d, %d, %d equals %d.\n', a,b,c,area)

**Function Script**

% Program Name: triangle.m

% Author: Juan Silva Last Modified: Mar. 26, 2018

% Description: This function will calculate the perimeter and area of a triangle. However, a restraint is applied to output an error the values entered do not meet the constraint.

% Function will add two resistors

% in parallel and give equivalent resistance

function[per, area] = triangles(a, b, c)

if((a + b) < c || (b + c) < a || (a + c) < b)

disp('Error. The sum of any two sides must be greater than the length of the third side.')

return

end

per = a + b + c;

s = per / 2;

area = sqrt(s \* (s - a) \* (s - b) \* (s - c));

end

end

**Results**

Enter value for side a: 3

Enter value for side b: 4

Enter value for side c: 5

The perimeter of a triangle with sides 3, 4, 5 equals 12.

The area of a triangle with sides 3, 4, 5 equals 6.

Enter value for side a: 2

Enter value for side b: 2

Enter value for side c: 5

Error. The sum of any two sides must be greater than the length of the third side.

Output argument "per" (and maybe others) not assigned during

call to "triangles".

**Problem Set 11-4:**

Suppose that we consider the series: 1 + ½ + ¼ + ... + 1/(2n-1) + ... and we would like to know how many terms we need to add until we reach the sum of 1.999. Use a while loop to determine how many terms are required.

**Main Script**

% Program Name: series.m

% Author: Juan Silva Last Modified: Feb. 28, 2018

% Description: This program prompts the user to enter three values and output the perimeter and area of a triangle.

clear, clc, close all

format short, format compact

%Problem 4

s = 1;

n = 0;

while true

s = s + (2^-(n + 1));

n = n + 1;

if s >= 1.999

break

end

fprintf('%f\n', s)

end

fprintf('The sum will equal 1.999 after %d terms.\n', n)

**Results**

1.500000

1.750000

1.875000

1.937500

1.968750

1.984375

1.992188

1.996094

1.998047

The sum will equal 1.999 after 10 terms.

**Problem Set 11-5:**

Write a function M-file called quad\_real that will only compute real roots of the quadratic equation. Your function should:  
–take inputs a, b, and c, the coefficients of the quadratic equation: ax2+ bx+ c = 0;  
–output the roots, say x1 and x2, only if the roots are real;  
–output an error message: “Sorry, the roots are complex.” if the roots are complex.

**Main Script**

% Program Name: quadReal\_TEST.m

% Author: Juan Silva Last Modified: Mar. 26, 2018

% Description: This program will compute the real roots of the quadratic equation. However, it will display an error if the roots are complex.

clear, clc

format short, format compact

a = input('Enter coeff for a: ');

b = input('Enter coeff for b: ');

c = input('Enter coeff for c: ');

[x1,x2] = quadReal(a,b,c);

if((b^2) < (4 \* a \* c))

disp('Sorry, the roots are complex.');

return

end

fprintf('(%d)x^2 + (%d)x + (%d) roots equal %.4f and %.4f\n', a, b, c, x1, x2)

**Function Script**

% Program Name: quadReal.m

% Author: Juan Silva Last Modified: Mar. 26, 2018

% Description: This function will calculate the capacitance between two parallel plates.

function [x1,x2] = quadReal(a, b, c)

x1 = (-b + sqrt(b^2 - 4\*a\*c)) / (2 \* a);

x2 = (-b - sqrt(b^2 - 4\*a\*c)) / (2 \* a);

end

**Results**

Enter coeff for a: 1

Enter coeff for b: 5

Enter coeff for c: 6

(1)x^2 + (5)x + (6) roots equal -2.00 and -3.00

**Problem Set 11-6:**

Write an M-file (called trig\_table) that generates 3 columns of data.  
•The 1st column should contain 10 x-values, linearly spaced, between 0 and 2π;  
•The 2nd column should contain sin(x);  
•The 3rd column should contain cos(x); The first column should contain 3 digits after the decimal point; the second and third columns should contain 5 digits beyond the decimal point. The spacing between columns should be 3 spaces. (No column headings are required.)

**Main Script**

% Program Name: trig\_table.m

% Author: Juan Silva Last Modified: Mar. 28, 2018

% Description: This program will generate 3 columns of data consisting of a column of theta, a column for sin(x), and a column cos(x).

x = linspace(0,2\*pi,10);

disp(' x cos(x) sin(x) ')

disp('--------------------------')

for i = 1: length(x)

fprintf('%5.3f %8.5f %8.5f\n', x(i), sin(i), cos(i))

end

**Results**

x cos(x) sin(x)

--------------------------

0.000 0.84147 0.54030

0.698 0.90930 -0.41615

1.396 0.14112 -0.98999

2.094 -0.75680 -0.65364

2.793 -0.95892 0.28366

3.491 -0.27942 0.96017

4.189 0.65699 0.75390

4.887 0.98936 -0.14550

5.585 0.41212 -0.91113

6.283 -0.54402 -0.83907

**Problem Set 11-7:**

Write a function M-file that will compute the equivalent resistance for a series or parallel combination of an arbitrary number of resistors. Your function should:

•accept as input an arbitrary number of resistor values, stored in a vector;

•ask the user to input ‘p’ if the resistors are in parallel;  
•ask the user to input ‘s’ if the resistors are in series;

•output the equivalent resistance for the series or parallel combination of resistors;

•output an error statement if the user input is not an ‘s’ or ‘p’.

Hint: you could use a for-loop (to go through all the resistors in the vector) and either if-statements or switch-statements to test for the different user inputs.

**Main Script**

% Program Name: resistance\_TEST.m

% Author: Juan Silva Last Modified: Mar. 27, 2018

% Description: This program prompts the user enter an arbitrary set of values and an argument to calculate the series / parallel resistance.

clear, clc, close

format long

size = input('Enter the size of array: ');

array = []; %initialize array

%Fill array

for n = 1: size

element = input('enter value: ');

array(n) = element;

end

x = input('Enter s for series or p for parallel: ', 's');

if x == 's'

series = sum(array);

fprintf('Series resistance equals %d?.\n', series)

elseif x == 'p'

par = prod(array) ./ sum(array);

fprintf('Parallel resistance equals %d?.\n', par)

else

disp('Argument error!')

end

**Results**

Enter the size of array: 5

enter value: 2

enter value: 4

enter value: 7

enter value: 5

enter value: 8

Enter s for series or p for parallel: s

Series resistance equals 26Ω.

Enter the size of array: 3

enter value: 1

enter value: 2

enter value: 3

Enter s for series or p for parallel: p

Parallel resistance equals 1Ω.