//AUTHOR: DAN CRISP

//CHAPTER 1 & 2

//PROBLEMS: 1.1, 1.2, 1.5, 1.10, 1.18, 2.4, 2.11, 2.12, 2.20, 2.21

#Q1 (1.1: Quadratic Roots)

quadroots.m

function [ z1, z2 ] = quadroots( a, b, c )

%quadroots: takes the coefficients for a given quadratic, and displays

%it's roots. example input: quadroots(1,2,-3)

clc

coeff = [a;b;c];

theRoots = roots(coeff);

disp('================')

disp('Quadratic Solver')

disp(' ')

disp('coefficients')

fprintf('\ta = %3.1f\n', coeff(1))

fprintf('\tb = %3.1f\n', coeff(2))

fprintf('\tc = %3.1f\n', coeff(3))

disp('roots')

fprintf('\tz1 = %4.2f\n', theRoots(1))

fprintf('\tz2 = %4.2f\n', theRoots(2))

end

OUTPUT:

================

Quadratic Solver

coefficients

a = 1.0

b = 2.0

c = -3.0

roots

z1 = -3.00

z2 = 1.00

#Q2 (1.2: Rolling Dice)

ThrowDice.m

function [ sum, vals ] = ThrowDice( num )

%ThrowDice: user throws num dice, function displays their values, and

%sum.

clc

%vals = [1:num];

sum = 0;

disp('=======================')

fprintf('\tThrowDice\n')

disp('=======================')

fprintf('\tDice(#)\t\tVal\n')

format = '\t%1.0f\t\t\t%1.0f\n';

for i=1:num

vals(i)=randi(6);

fprintf(format,i,vals(i));

sum = sum + vals(i);

end

fprintf('\tSUM\t\t\t%2.0f\n',sum)

%return vals, sum;

end

OUTPUT: (for >>ThrowDice(6);)

=======================

ThrowDice

=======================

Dice(#) Val

1 6

2 6

3 1

4 6

5 6

6 3

SUM 28

#Q3 (1.5: Gravitational Force)

gforce.m

function [ f ] = gforce( m1 , m2 , dist )

clc

% Gravitational Constant

G = 6.67300\*10^-11;

% Force of Gravity

f = G\*m1\*m2/(dist^2);

% Display

fprintf('\tG\t\t= %4.12f\n',G)

fprintf('\tMass#1\t= %4.0f\n',m1)

fprintf('\tMass#2\t= %4.0f\n',m2)

fprintf('\tdist\t= %4.1f\n',dist)

fprintf('\tFORCE\t= %4.12f\n\n\n',f)

end

OUTPUT: (for >>gforce(60,80,2);)

Force of Gravity

G = 0.000000000067

Mass#1 = 80

Mass#2 = 60

dist = 2.0

FORCE = 0.000000080076

#Q4 (1.10: Elastic Collisions in One Dimension)

Collide.m

function [ output\_args ] = Collide( m1, m2, v1, v2 )

%Collide: Given both mass and initial velocity of two objects,

% final velocities are given for an elastic collision.

% Clear Console

clc

% Calculation

v1f = ((m1-m2)/(m1+m2))\*v1+((2\*m2)/(m1+m2))\*v2;

v2f = ((2\*m1)/(m1+m2))\*v1+((m2-m1)/(m1+m2))\*v2;

% Display

disp('=======================================')

fprintf('\t\t\tElastic Collision\n')

disp('=======================================')

format = '%1.0f\t\t%4.0f\t%4.0f\t\t%4.0f\n';

fprintf('#\t\tMass\tInitial V\tFinal V\n')

fprintf(format,1,m1,v1,v1f)

fprintf(format,2,m2,v2,v2f)

end

OUTPUT: (for >>Collide(5,3,2,-4));)

=======================================

Elastic Collision

=======================================

# Mass Initial V Final V

1 5 2 -3

2 3 -4 4

#Q5 (1.18: Compound Interest)

CompInterest.m

function [ output\_args ] = CompInterest( P, r, k, Ny )

%CompInterest Analyze the limit for compound interest

% Using Value of interest-bearing investment of principle, P, after Ny

% years, 'k' compound frequency per year, and 'r' interest rate.

% Clear Console

clc

% Calculation

V = P\*(1+r/k)^(k\*Ny);

% Display

disp('=======================================')

fprintf('\t\t\tCompInterest\n')

disp('=======================================')

fprintf('\tPrinciple\t= $%4.2f\n',P)

fprintf('\tRate\t\t= %4.2f\n',r)

fprintf('\tComp.Freq.\t= %4.0f\n',k)

fprintf('\tYears\t\t= %4.1f\n',Ny)

fprintf('\tValue\t\t= $%4.2f\n\n\n',V)

end

OUTPUT: (for >>CompInterest(1,1,1000000000,1))

=======================================

CompInterest

=======================================

Principle = $1.00

Rate = 1.00

Comp.Freq. = 1000000000

Years = 1.0

Value = $2.72

After one year, the value is $2.72. This is very close to the mathimatical constant 'e' used for the base of natural logorithms.

#Q6 (2.4: Sum Next 10)

sum10.m

%sum10 Sums user specified number with 9 successive values.

% Clear Console

clc

% Input

prompt = 'Please enter a number: ';

x = round(input(prompt));

% Retrieve next 9 numbers, and sum them

sum = 0;

for i = 1:10

sumnums{i} = x + i - 1;

sum = sum + sumnums{i};

end

% Output

disp('=======================================')

fprintf('\t\t\tsum10\n')

disp('=======================================')

fprintf('The sum of %2.0f and the next 9 numbers is %2.0f\n\n',x,sum)

OUTPUT: (for input of 5.5)

Please enter a number: 5.5

=======================================

sum10

=======================================

The sum of 6 and the next 9 numbers is 105

#Q7 (2.11: Password)

password.m

%password.m generates a password based on information given by user

% Clear Variables, and Screen

clear; clc;

% Request user's info

first = input('Please enter your name: ','s');

clc

fprintf('NAME:\n%s\n\n',first);

mid = input('Please enter your middle initial: ','s');

clc

fprintf('NAME:\n%s\t%s\n\n',first, mid);

last = input('Please enter your last name: ','s');

clc

fprintf('NAME:\n%s\t%s\t%s\n\n',first, mid, last);

phone = num2str(input('Please enter your 10 digit phone number: '));

clc

fprintf('NAME:\n%s\t%s\t%s\n\n',first, mid, last);

fprintf('NUMBER:\n%s\n\n', phone);

% Generate Password Segments

F = lower(first(1));

M = lower(mid(1));

L = lower(last(1:2));

to = numel(phone);

from = to - 2;

P = phone(from:to);

% Concatenate Segments, Display Password

fprintf('PASSWORD:\n%s%s%s%s\n\n',F,M,L,P)

OUTPUT:

NAME:

Daniel B Crisp

NUMBER:

6301234567

PASSWORD:

dbcr567

#Q8 (2.12: Take Order)

TakeOrder.m

%TakeOrder.m greets customer, takes their name and their order.

% Clear Screen

clc;

name = input('Please enter your name: ','s');

fprintf('Good evening, %s! The kitchen is cooking some good stuff in the back.\n\n',name)

apps = input('How may appetizers will I be getting for you today?\n\n');

salads = input('Anyone up for salads? How many can I put yall down for?\n\n');

twinkies = input('And how many fried twinkies?\n\n');

total = apps + salads + twinkies;

fprintf('\nFantastic. I have you down for %2.0f items total.\n\n',total)

OUTPUT:

Please enter your name: Dan

Good evening, Dan! The kitchen is cooking some good stuff in the back.

How may appetizers will I be getting for you today?

2

Anyone up for salads? How many can I put yall down for?

4

And how many fried twinkies?

1

Fantastic. I have you down for 7 items total.

#Q9 (2.20: Furniture Sales)

Furniture.m

% Furniture.m Given 4 different sets of furnature, and values for number

% sold, their cost to both store and customer, calculate these values:

% Total sold, Total cost, and total profit.

% Clear Screen

clc;

% Cost of each set

Cs = [199, 268, 500, 670];

% Price of each set

Ps = [398, 598, 798, 998];

% Number of each set sold

Ns = [35, 25, 20, 10];

% Total sets sold, total revenue, and total profit

totalSold = 0;

totalRevenue = 0;

totalCost = 0;

totalProfit = 0;

for i = 1:4

totalSold = totalSold + Ns(i);

totalRevenue = totalRevenue + Ps(i)\*Ns(i);

totalCost = totalCost + Cs(i)\*Ns(i);

Prof(i) = (Ps(i)-Cs(i))\*Ns(i);

totalProfit = totalProfit + Prof(i);

Prof(i) = totalProfit;

end

% Totals appended to arrays

Ns(5) = totalSold;

Ps(5) = totalRevenue;

Cs(5) = totalCost;

Prof(5) = totalProfit;

% Data Table

dataTable = [Ns; Ps; Cs; Prof];

% Display

fprintf('\t\tSetA\tSetB\tSetC\tSetD\tTOTALS\n');

fprintf('# Sold\t%3.0f\t\t%3.0f\t\t%3.0f\t\t%3.0f\t\t%5.0f\n',Ns(1),Ns(2),Ns(3),Ns(4),Ns(5));

fprintf('Revenue\t%3.0f\t\t%3.0f\t\t%3.0f\t\t%3.0f\t\t%5.0f\n',Ps(1),Ps(2),Ps(3),Ps(4),Ps(5));

fprintf('Cost\t%3.0f\t\t%3.0f\t\t%3.0f\t\t%3.0f\t\t%5.0f\n',Cs(1),Cs(2),Cs(3),Cs(4),Cs(5));

fprintf('Profit\t%3.0f\t%3.0f\t%3.0f\t%3.0f\t$%5.0f\n',Prof(1),Prof(2),Prof(3),Prof(4),Prof(5));

OUTPUT:

SetA SetB SetC SetD TOTALS

# Sold 35 25 20 10 90

Revenue 398 598 798 998 54820

Cost 199 268 500 670 30365

Profit 6965 15215 21175 24455 $24455

#Q10 (2.21: Rental Receipts)

% rentals.m Using car rental data for three locations, calculate and

% display the total number of rental days at each location and overall, then do the same for revenue.

% Clear Screen

clc;

% data

Cars = ['compacts', 'full-sized', 'vans', 'suvs'];

perDay = [25, 38, 53, 72];

daysAtAirport = [250, 150, 180, 86];

daysAtCampus = [160, 44, 60, 20];

daysAtElkhart = [210, 112, 120, 78];

% Total number of rental days at each location

totalDaysAirport = 0;

totalRevAirport = 0;

totalDaysCampus = 0;

totalRevCampus = 0;

totalDaysElkhart = 0;

totalRevElkhart = 0;

% process

for i = 1:4

totalDaysAirport = totalDaysAirport + daysAtAirport(i);

totalRevAirport = totalRevAirport + perDay(i)\*daysAtAirport(i);

totalDaysCampus = totalDaysCampus + daysAtCampus(i);

totalRevCampus = totalRevCampus + perDay(i)\*daysAtCampus(i);

totalDaysElkhart = totalDaysElkhart + daysAtElkhart(i);

totalRevElkhart = totalRevElkhart + perDay(i)\*daysAtElkhart(i);

end

totalDays = totalDaysAirport + totalDaysCampus + totalDaysElkhart;

totalRevenue = totalRevAirport + totalRevCampus + totalRevElkhart;

% display

fprintf('\nTotal number of rental days at each location:\n');

fprintf('Airport:\t%2.0f\n',totalDaysAirport);

fprintf('Campus:\t\t%2.0f\n',totalDaysCampus);

fprintf('Elkhart:\t%2.0f\n\n',totalDaysElkhart);

fprintf('Total number of rental days throughout the company: %3.0f\n\n',totalDays);

fprintf('\nTotal amount of revenue from each location:\n');

fprintf('Airport:\t$%2.0f\n',totalRevAirport);

fprintf('Campus:\t\t$%2.0f\n',totalRevCampus);

fprintf('Elkhart:\t$%2.0f\n\n',totalRevElkhart);

fprintf('Total amount of revenue earned throughout the company: $%3.0f\n\n',totalRevenue);

OUTPUT:

Total number of rental days at each location:

Airport: 666

Campus: 284

Elkhart: 520

Total number of rental days throughout the company: 1470

Total amount of revenue from each location:

Airport: $27682

Campus: $10292

Elkhart: $21482

Total amount of revenue earned throughout the company: $59456