Chapter 7
Solved Problems

Problem 1

User-defined function:

```
function mpg=kmlTOmpg(kml)
%mgTOkm converts fuel efficiency from kilometers per liter
% to miles per gallon.
mpg=kml*3.7854/1.60934;
```

1.*a*
Command Window:

```
>> Part_a=kmlTOmpg(9)
Part_a =
   21.1693
```

1.*b*
Command Window:

```
>> Part_b=kmlTOmpg(14)
Part_b =
   32.9300
```

Problem 2

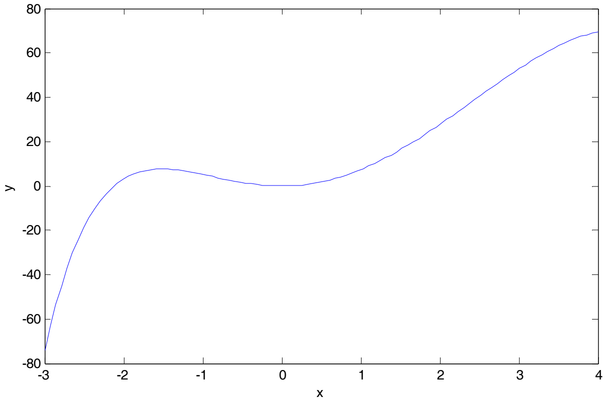
User-defined function:

```
function y=HW7_2(x)
y=-0.2*x.^4+exp(-0.5*x).*x.^3+7*x.^2;
```

Command Window:

```
Part (a)
yaa =
  -18.5991
yab =
   52.8245
Part (b)
```

Figure:



Problem 3

User-defined function:

```
function [cm,kg]=STtoSI(in,lb)
cm=in*2.54;
kg=lb/2.205;
```

3.*a*

Command Window:

```
>> [h,w]=STtoSI(68,175)
h =
  172.7200
w =
   79.3651
```

Problem 4

User-defined function:

```
function mps=mphTOmets(mph)
%mphTOmets converts speed from mile per hour per hour to
meter per second.
mps=mph*1609.344/3600;
```

Command Window:

```
>> mps=mphTOmets(55)
mps =
   24.5872
```

Problem 5

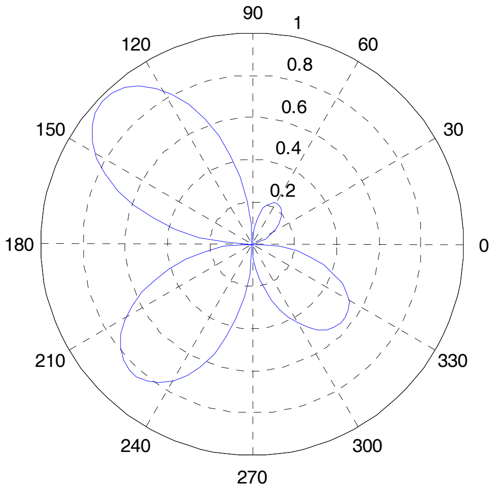
User-defined function:

```
function r=HW7_5(th)
r=2*cos(1*th).*sin(th).*sin(th/4);
```

Command Window:

```
>> mps=mphTOmets(55)
mps =
   24.5872
```

Figure:



Problem 6

User-defined function:

```
function Area = triangle(a,b,c)
S=(a+b+c)/2;
Area=sqrt(S*(S-a)*(S-b)*(S-c));
```

Command Window:

```
>> AreaPart_a=triangle(3,8,10)
AreaPart_a =
    9.9216
>> AreaPart_b=triangle(7,7,5)
AreaPart_b =
   16.3459
```

Problem 7

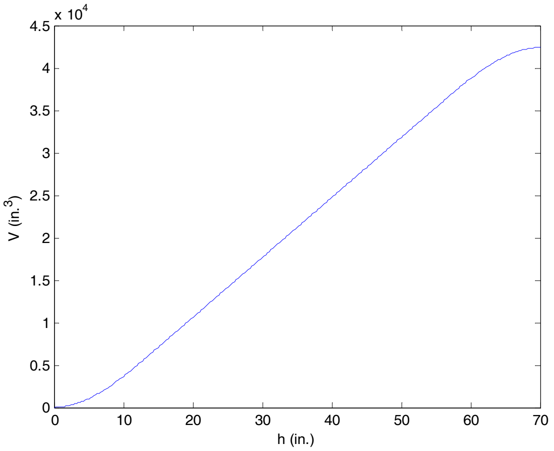
User-defined function:

```
function V=VolFuel(h)
r=15; H=40;
if h <= r
    V=pi*h^2*(3*r-h)/3;
elseif h > r & h <= (r+H)
    V=2*pi*r^3/3+pi*r^2*(h-r);
elseif h > (H+r) & h <= (H+2*r)
    V=4*pi*r^3/3+pi*r^2*H-pi*(H+2*r-h)^2*(r-H+h)/3;
end
```

Script file:

```
clear,clc
h=0:0.25:70;
n=length(h);
for i=1:n
V(i)=VolFuel(h(i));
end
plot(h,V)
xlabel('h (in.)')
ylabel('V (in.^3)')
```

Figure:



Problem 8

Script file:

```
clear, clc
W=@ (r,d,t,ga) ga*pi^2*(2*r+d)*d*t;
Wa=W(0.35,0.12,0.02,0.696)
```

Command Window:

```
Wa =
    0.0135
```

Problem 9

User-defined function:

```
function M = invest(S,r,N)
ra=r/1200;
M=S*ra/((1+ra)^(12*N)-1);
```

Command Window:

```
>> MonthlyDeposit=invest(25000,4.24,10)
MonthlyDeposit =
  167.6409
```

Problem 10

User-defined function:

```
function HI = HeatIn(T,R)
C1=-42.379; C2=2.04901523; C3=10.14333127;
C4=-0.22475541; C5=-6.83783E-3; C6=-5.481717E-2;
C7=1.22874E-3; C8=8.5282E-4; C9=-1.99E-6;
HI=C1+C2*T+C3*R+C4*T*R+C5*T^2+C6*R^2+C7*T^2*R+C8*T*R^2+C9*R^
2*T^2;
HI=round(HI);
end
```

Command Window:

```
>> HI_Part_a = HeatIn(95,80)
HI_Part_a =
   134
>> HI_Part_b = HeatIn(100,100)
HI_Part_b =
   195
```

Problem 11

User-defined function:

```
function BFP = BodyFat(w,h,age,gen)
% BodyFat calculates body fat procentage.
% Input variables:
% w  Weight in lb.
% h  Height in in.
% age  Age.
% gender  Gender (Enter 1 for male, 0 for female).
% Output variable:
% BFP  Body fat procentage.
BMI=703*w/h^2;
BFP=1.2*BMI+0.23*age-10.8*gen-0.54;
```

Command Window:

```
>> BFP_Part_a = BodyFat(220,74,35,1)
BFP_Part_a =
   30.6019
>> BFP_Part_b = BodyFat(135,67,22,0)
BFP_Part_b =
   29.8900
```

Problem 12

User-defined function:

```
function av = GPA( g,h )
n=length(h);
for i=1:n
    gN(i)=LetTONum(g(i));
end
TotCH=sum(h);
av=sum(gN.*h)/TotCH;


function N=LetTONum(L)
switch L
    case 'A'
        N=4;
    case 'B'
        N=3;
    case 'C'
        N=2;
    case 'D'
        N=1;
    case 'E'
        N=0;
end
```

Script File:

```
Grades=['BACEABDB'];
CredHr=[3 4 3 4 3 4 3 2];
Grade_Point_Ave=GPA(Grades,CredHr)
```

Command Window:

```
Grade_Point_Ave =
    2.4615
```

Problem 13

User-defined function:

```
function y=fact(x)
if x==0
    y=1;
elseif x<0
    y='Error';
    disp('ERROR: Input argument must be a positive integer')
elseif round(x)~=x
    y='Error';
    disp('ERROR: Input argument must be a positive integer')
else
    n=x;
    y=1;
    for i=1:n
        y=y*i;
    end
end
```

Command Window:

```
>> y_Part_a=fact(12)
y_Part_a =
   479001600
>> y_Part_b=fact(0)
y_Part_b =
     1
>> y_Part_c=fact(-7)
ERROR: Input argument must be a positive integer
y_Part_c =
Error
>> y_Part_d=fact(6.7)
ERROR: Input argument must be a positive integer
y_Part_d =
Error
```

Problem 14

User-defined function:

```
function V=vector(A,B)
V=B-A;
```

Command Window:

```
>> V_Part_a=vector([0.5 1.8], [-3 16])
V_Part_a =
   -3.5000   14.2000
>> V_Part_b=vector([-8.4 3.5 -2.2], [5 -4.6 15])
V_Part_b =
   13.4000   -8.1000   17.2000
```

Problem 15

User-defined function:

```
function D = dotpro(u,v)
n=length(u);
D=0;
for i=1:n
    D=D+u(i)*v(i);
end
```

Command Window:

```
>> D_Part_a = dotpro([3 11],[14 -7.3])
D_Part_a =
  -38.3000
>> D_Part_b = dotpro([-6 14.2 3],[6.3 -8 -5.6])
D_Part_b =
 -168.2000
```

Problem 16

User-defined function:

```
function n=unitvec(A,B)
C=B-A;
d=sqrt(sum(C.^2));
n=C/d;
```

Command Window:

```
>> n_Part_a = unitvec([1.2 3.5],[12 15])
n_Part_a =
    0.6846    0.7289
>> n_Part_b = unitvec([-10 -4 2.5],[-13 6 -5])
n_Part_b =
   -0.2334    0.7779   -0.5834
```

Problem 17

User-defined function:

```
function w = crosspro(u,v)
n=length(u);
if n == 2
    u(3)=0;
    v(3)=0;
end
w(1)=u(2)*v(3)-u(3)*v(2);
w(2)=u(3)*v(1)-u(1)*v(3);
w(3)=u(1)*v(2)-u(2)*v(1);
```

Command Window:

```
>> w_Part_a = crosspro([3 11],[14 -7.3])
w_Part_a =
         0         0 -175.9000
>> w_Part_b = crosspro([-6 14.2 3],[6.3 -8 -5.6])
w_Part_b =
  -55.5200  -14.7000  -41.4600
```

Problem 18

User-defined function:

```
function Area = TriArea(A,B,C)
VAB=vector(A,B);
VAC=vector(A,C);
w = crosspro(VAB,VAC);
D = dotpro(w,w);
Area=sqrt(D)/2;

function V=vector(A,B)
V=B-A;

function w = crosspro(u,v)
n=length(u);
if n == 2
    u(3)=0;
    v(3)=0;
end
w(1)=u(2)*v(3)-u(3)*v(2);
w(2)=u(3)*v(1)-u(1)*v(3);
w(3)=u(1)*v(2)-u(2)*v(1);

function D = dotpro(u,v)
n=length(u);
D=0;
for i=1:n
    D=D+u(i)*v(i);
end
```

Command Window:

```
>> Area = TriArea([1 2],[10 3],[6 11])
Area =
    38
>> Area = TriArea([-1.5 -4.2 -3],[-5.1 6.3 2],[12.1 0 -
0.5])
Area =
   87.8853
```

Problem 19

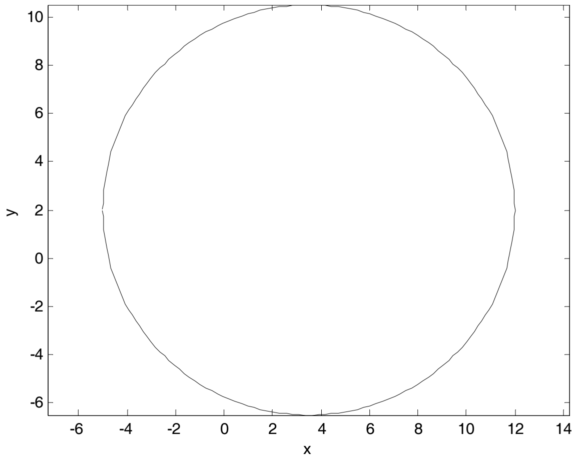
User-defined function:

```
function  circleplot(x,y,R)
th=linspace(0,pi,100);
xp=x+R*cos(th);
ypp=y+R*sin(th);
ypn=y-R*sin(th);;
plot(xp,ypp,'k',xp,ypn,'k')
axis equal
xlabel('x')
ylabel('y')
```

Part a Command Window:

```
>> circleplot(3.5,2,8.5)
```

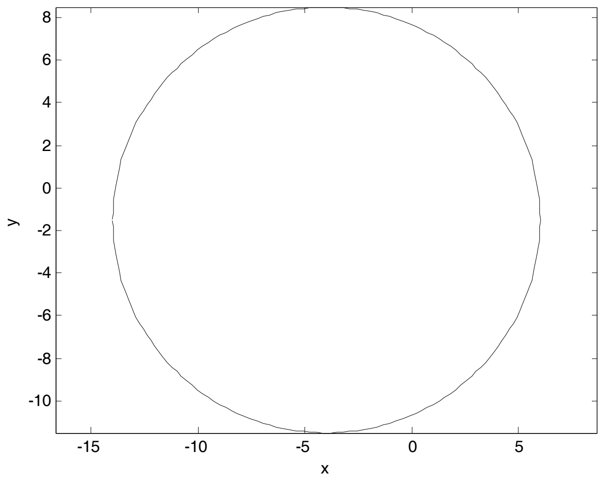
Part a Figure:



Part b Command Window:

```
>> >> circleplot(-4,-1.5,10)
```

Part b Figure:



Problem 20

The equation of the circle can be rewritten as:

$$-2xa-2yb+a^2+b^2-r^2 = -x^2-y^2$$

Substituting the three points in the equation gives a system of three linear equations for the unknowns: a, b, and $a^2+b^2-r^2$. Once the system is solved a and b are known, and r can be determined since $a^2+b^2-r^2$ is known.

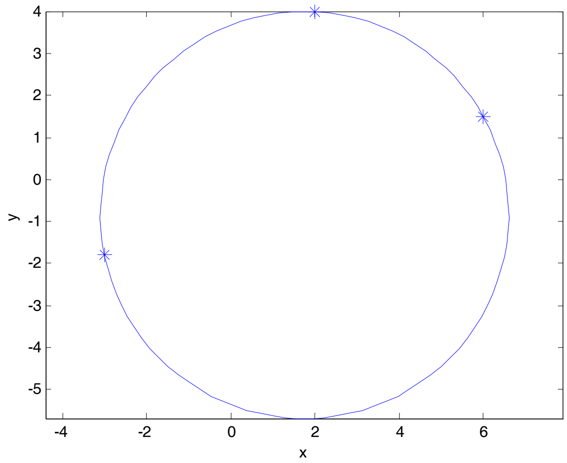
User-defined function:

```
function cirpnts(P)
A=[-2*P(1,1) -2*P(1,2) 1; -2*P(2,1) -2*P(2,2) 1; -2*P(3,1) -
2*P(3,2) 1];
B=[-(P(1,1)^2+P(1,2)^2); -(P(2,1)^2+P(2,2)^2); -
(P(3,1)^2+P(3,2)^2)];
C = A\B;
a=C(1);
b=C(2);
r=sqrt(C(1)^2+C(2)^2-C(3));
circleplot(a,b,r)
axis equal
xlabel('x'), ylabel('y')
hold on
plot(P(:,1),P(:,2),'*','markersize',10)
hold off

function circleplot(x,y,R)
th=linspace(0,2*pi,100);
xp=x+R*sin(th);
yp=y+R*cos(th);
plot(xp,yp)
```

Command Window:

```
>> Points=[6 1.5; 2 4; -3 -1.8];
>> cirpnts(Points)
```

Figure:



Problem 21

User-defined function:

```
function [r th]=AddVelPol(r1,th1,r2,th2)
x=r1*cosd(th1)+r2*cosd(th2);
y=r1*sind(th1)+r2*sind(th2);
r=sqrt(x^2+y^2);
th=acosd(x/r);
if y<0
    th=360-th;
end
```

Command Window:

```
>> [ra tha]=AddVelPol(5,23,12,40)
ra =
   16.8451
tha =
   35.0215
>> [rb thb]=AddVelPol(6,80,15,125)
rb =
   19.7048
thb =
  112.5663
```

Problem 22

User-defined function:

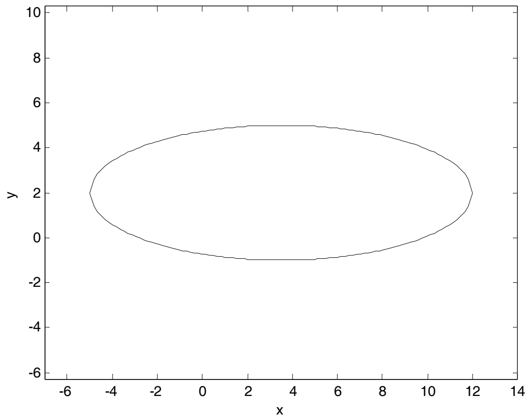
```
function ellipseplot(xc,yc,a,b)
x=linspace(-a,a,100);
y=sqrt(b^2*(1-x.^2/a^2));
xp=x+xc;
ypp=y+yc;
ypm=-y+yc;
plot(xp,ypp,'k',xp,ypm,'k')
%axis square
axis([xc-a-2,xc+a+2,yc-b-2,yc+b+2])
axis equal

xlabel('x'), ylabel('y')
```

Command Window:

22.*a*

```
ellipseplot(3.5,2,8.5,3)
```

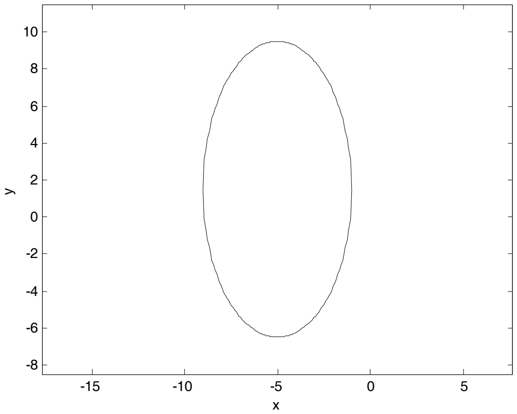
Figure:



22.*b*

```
>> ellipseplot(-5,1.5,4,8)
```

Figure:



Problem 23

User-defined function:

```
function pr=prime(m,n)
% prime determines all the prime numbers between m and n.
% Input argument:
% m  An interger.
% n  An interger (n>m).
% Output argument:
% pr  A vector whose elements are the prime numbers between 1
and n.
if n<=0
    pr='Error';
    disp('ERROR: Input argument must be a positive integer')
elseif round(n)~=n | round(m)~=m
    pr='Error';
    disp('ERROR: Input argument must be positive integer')
elseif n <= m
    pr='Error';
    disp('ERROR: n must be greater than m')
else
    k=1;
    for i=m:n
         c=0;
         for j=2:i-1
             if rem(i,j)==0
                 c=1;
                 break
             end
         end
         if c==0
             pr(k)=i;
             k=k+1;
         end
     end
end
```

Command Window:

23.*a*

```
>> pr=prime(12,80)
pr =
  Columns 1 through 9
    13    17    19    23    29    31    37    41    43
  Columns 10 through 17
    47    53    59    61    67    71    73    79
```

23.*b*

```
>> pr=prime(21,63.5)
ERROR: Input argument must be positive integer
pr =
Error
```

23.*c*

```
>> pr=prime(100,200)
pr =
  Columns 1 through 9
   101   103   107   109   113   127   131   137   139
  Columns 10 through 18
   149   151   157   163   167   173   179   181   191
  Columns 19 through 21
   193   197   199
```

23.*d*

```
>> pr=prime(90,50)
ERROR: n must be greater than m
pr =
Error
```

Problem 24

User-defined function:

```
function GM=Geomean(x)
n=length(x);
GM=x(1)*x(2);
if n>2
    for i=3:n
        GM=GM*x(i);
    end
end
GM=GM^(1/n);
```

Script File:

```
clear
x=[1.38 1.76 1.17 0.79 1.42 0.64 1.2 1.06 0.83 1.18];
GM=Geomean(x)
```

Command Window:

```
GM =
    1.09780
```

Answer: Average return is 9.78% per year.

Problem 25

User-defined function:

```
function [theta, radius]=CartesianToPolar(x,y)
radius= sqrt(x^2+y^2);
theta=acos(abs(x)/radius)*180/pi;
if (x<0)&(y>0)
    theta=180-theta;
end
if (x>0)&(y<0)
    theta=-theta;
end
if (x<=0)&(y<0)
    theta=theta-180;
end
```

Command Window:

```
>> [th_a, radius_a]=CartesianToPolar(14,9)
th_a =
   32.7352
radius_a =
   16.6433
>> [th_b, radius_b]=CartesianToPolar(-11,-20)
th_b =
 -118.8108
radius_b =
   22.8254
>> [th_c, radius_c]=CartesianToPolar(-15,4)
th_c =
  165.0686
radius_c =
   15.5242
>> [th_d, radius_d]=CartesianToPolar(13.5,-23.5)
th_d =
  -60.1240
radius_d =
   27.1017
```

Problem 26

User-defined function:

```
function y=downsort(x)
n=length(x);
for i=1:n-1
    T=x(i);
    for j=i+1:n
        if x(j)>T;
        x(i)=x(j);
        x(j)=T;
        T=x(i);
        end
    end
end
 y=x;
```

Command Window:

```
A =
   -23      0     28    -10      5    -17     15    -15      0
12     24     28      3    -22
>> Asorted=downsort(A)
Asorted =
    28     28     24     15     12      5      3      0      0
-10    -15    -17    -22    -23
>>
```

Problem 27

User-defined function:

```
function B = matrixsort(A)
[r,c]=size(A);
for i=1:r
    ja=(i-1)*c+1;
    jb=i*c;
Avector(ja:jb)=A(i,:);
end
AvectorSorted=downsort(Avector);
for i=1:r
    ja=(i-1)*c+1;
    jb=i*c;
B(i,1:c)=AvectorSorted(ja:jb);
end

function y=downsort(x)
n=length(x);
for i=1:n-1
    T=x(i);
    for j=i+1:n
        if x(j)>T;
        x(i)=x(j);
        x(j)=T;
        T=x(i);
        end
    end
end
 y=x;
```

Command Window:

```
>> A=randi([-30 30],4,7)
A =
   -21    19   -19    -9    25    -7     2
   -15   -16   -15    20   -13     4    17
    21    26     7     5    16   -26    26
   -15    -9    -2     3    15   -27   -23
```

```
>> B = matrixsort(A)
B =
    26    26    25    21    20    19    17
    16    15     7     5     4     3     2
    -2    -7    -9    -9   -13   -15   -15
   -15   -16   -19   -21   -23   -26   -27
>>
```

Problem 28

User-defined function:

```
function d3=det3by3(A)
S1=A(2:3,2:3);
S2=A(2:3,[1 3]);
S3=A(2:3,1:2);
d3=A(1,1)*det2by2(S1)-A(1,2)*det2by2(S2)+A(1,3)*det2by2(S3);

function d2=det2by2(B)
d2=B(1,1)*B(2,2)-B(1,2)*B(2,1);
```

Command Window:

```
>> A=[1 3 2; 6 5 4; 7 8 9];
>> det_Part_a=det3by3(A)
det_Part_a =
   -39
>> B=[-2.5 7 1; 5 -3 -2.6; 4 2 -1];
>> det_Part_b=det3by3(B)
det_Part_b =
  -36.3000
```

Problem 29

User-defined function:

```
function [Smax,Smin]=princstress(Sxx,Syy,Sxy)
AVE=(Sxx+Syy)/2;
R=sqrt(((Sxx-Syy)/2)^2+Sxy^2);
Smax=AVE+R;
Smin=AVE-R;
```

Command Window:

29.*a*

```
>>  [S1a,S2a]=princstress(-190,145,110)
S1a =
  177.8902
S2a =
 -222.8902
```

Answer: The principal stresses are: 177.8902 MPa and -222.8902 MPa.

29.*b*

```
>> [S1a,S2a]=princstress(14,-15,8)
S1a =
   16.0605
S2a =
  -17.0605
```

Answer: The principal stresses are: 16.0605 ksi and -17.0605ksi.

Problem 30

User-defined function:

```
function [ Td, RH ] = DewptRhum(T, Tw, BP)
es=6.112*exp(17.67*T/(T+243.5));
ew=6.112*exp(17.67*Tw/(Tw+243.5));
e=ew-BP*(T-Tw)*0.00066*(1+0.00115*Tw);
RH=100*e/es;
RH=round(RH*10)/10;
Td=243.5*log(e/6.112)/(17.67-log(e/6.112));
Td=round(Td*10)/10;
end
```

Command Window:

```
>> [ Td_Part_a, RH_Part_a ] = DewptRhum(25, 19, 985)
Td_Part_a =
   15.8000
RH_Part_a =
   56.7000
>>
>> [ Td_Part_b, RH_Part_b ] = DewptRhum(36, 31, 1020)
Td_Part_b =
   29.6000
RH_Part_b =
   69.7000
```

Problem 31

User-defined function:

```
function g=fgrade(R)
[r c]=size(R);
for i= 1:r
    [HWmin HWn]=min(R(i,1:6));
    HWav=(sum(R(i,1:6))-R(i,HWn))/5;
    HW=HWav*15/10;
    MTav=sum(R(i,7:9))/3;
    if MTav > R(i,10)
        MT=MTav*85/100;
        gr(i)=HW+MT;
    else
        MT=sum(R(i,7:9))*15/100;
        F=R(i,10)*40/100;
        gr(i)=HW+MT+F;
    end
end
g=gr';
```

Script file:

```
clear,clc
disp('Part a')
Ra=[7 10 6 9 10 9 91 71 81 88]
ga=fgrade(Ra)
disp('Part b')
Rb=[7 10 6 9 10 9 91 71 81 88
    5 5 6 1 8 6 59 72 66 59
    6 8 10 4 5 9 72 78 84 78
    7 7 8 8 9 8 83 82 81 84]
gb=fgrade(Rb)
```

Command Window:

```
Part a
Ra =
     7    10     6     9    10     9    91    71    81
88
ga =
   85.1500
Part b
Rb =
     7    10     6     9    10     9    91    71    81
88
     5     5     6     1     8     6    59    72    66
59
     6     8    10     4     5     9    72    78    84
78
     7     7     8     8     9     8    83    82    81
84
gb =
   85.1500
   64.8167
   77.7000
   82.5000
>>
```

Problem 32

User-defined function:

```
function x=lotto(a,b,n)
% loto selects n numbers (all different) from the domain a,b.
% x is a vector with the n numbers.
x(1)=randi([a b]);
for p=2:n
    x(p)=randi([a b]);
    r=0;
    while r==0
        r=1;
        for k=1:p-1
            if x(k)==x(p)
               x(p)=randi([a b]);
               r=0;
               break
            end
        end
    end
end
```

Command Window:

```
xa =
    10     36     16     39     41     45     27
>> xb=lotto(50,65,8)
xb =
    51     53     64     52     63     58     65     57
>> xc=lotto(-25,-2,9)
xc =
   -23     -2    -25     -7     -6     -5    -16    -19    -15
```

Problem 33

User-defined function:

```
function y=fifthroot(P)
imax=50;
Emax=0.00001;
x=P;
for i=1:imax
    y=x-(x^5-P)/(5*x^4);
    E=abs((y-x)/x);
    x=y;
    if E<Emax
        break
    end
end
if E>Emax
    y='ERROR';
    disp('Solution was not obtained in 50 iterations')
end
```

Command Window:

```
>> ya=fifthroot(120)
ya =
    2.6052
>> yb=fifthroot(16807)
yb =
    7.0000
>> yc=fifthroot(-15)
yc =
   -1.7188
>>
```

Problem 34

User-defined function:

```
function yc=centroidT(w,h,t,d)
yc=(w*t*(h-t/2)+d*(h-t)^2/2)/(w*t+d*(h-t));
```

Command Window:

```
>> yc=centroidT(240,380,60,42)
yc =
  258.2759
```

Problem 35

User-defined function:

```
function Ixc=IxcTBeam(w,h,t,d)
yc=centroidT(w,h,t,d);
A=d*(h-t)^3/12+d*(h-t)*(yc-(h-t)/2)^2;
B=w*t^3/12+w*t*((h-t/2)-yc)^2;
Ixc=A+B;

function yc=centroidT(w,h,t,d)
yc=(d*(h-t)*(h-t)/2+w*t*(h-0.5*t))/(d*(h-t)+w*t);
```

Command Window:

```
>> Ixc=IxcTBeam(240,380,60,42)
Ixc =
  3.6997e+008
```

Problem 36

User-defined function:

```
function RV=lowpass(R,C,w)
RV=1./sqrt(1+(w*R*C).^2);
```

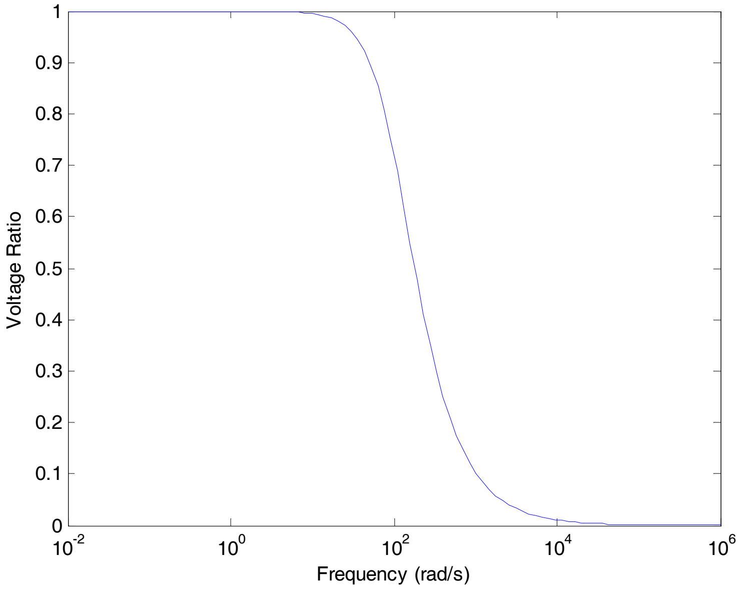
Script File:

```
R=input('Enter the value of R in ohms  ');
C=input('Enter the value of C in Farad  ');
n=linspace(-2,6,100);
w=10.^n;
RV=lowpass(R,C,w);
semilogx(w,RV)
xlabel('Frequency (rad/s)')
ylabel('Voltage Ratio')
```

Command Window:

```
Enter the value of R in ohms  1200
Enter the value of C in Farad  8e-6
```

Figure:



Problem 37

User-defined function:

```
function RV=bandpass(R,C,L,w)
D=sqrt((1-w.^2*L*C).^2+(w*R*C).^2);
RV=w*R*C./D;
```

Script File:

```
R=input('Enter the value of R in ohms  ');
C=input('Enter the value of C in Farad  ');
L=input('Enter the value of L in henry  ');
n=linspace(-2,7,100);
w=10.^n;
RV=bandpass(R,C,L,w);
semilogx(w,RV)
xlabel('Frequency (rad/s)')
ylabel('Voltage Ratio')
```

Command Window:

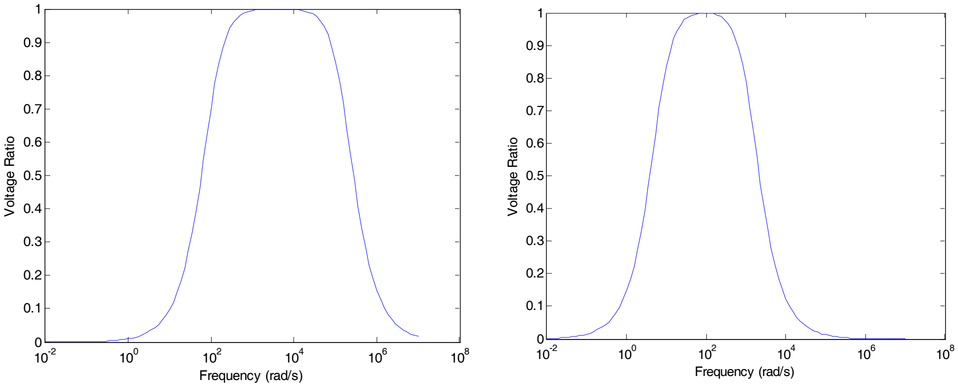
37.*a*

```
Enter the value of R in ohms  1100
Enter the value of C in Farad  9e-6
Enter the value of L in henry  7e-3
```

37.*b*

```
Enter the value of R in ohms  500
Enter the value of C in Farad  300e-6
Enter the value of L in henry  400e-3
```

Figure:



Problem 38

User-defined function:

```
function dydx = FunDer( Fun,x )
h=x/10;
fm2=Fun(x-2*h);
fm1=Fun(x-1*h);
fp2=Fun(x+2*h);
fp1=Fun(x+1*h);
dydx=(fm2-8*fm1+8*fp1-fp2)/(12*h);
end
```

Script File:

```
clear, clc
% Part A
disp('Part a')
Fa= @ (x) x^2*exp(x);
dydxA = FunDer( Fa,0.25 )
% Part B
disp('Part b')
Fb= @ (x) 2^x/x;
dydxB = FunDer( Fb,2 )
```

Commans Window:

```
Part a
dydxA =
    0.7223
Part b
dydxB =
    0.3864
```

Problem 39

User-defined function:

```
function [xr,yr] = rotation(x,y,q)
xr=cosd(q)*x-sind(q)*y;
yr=sind(q)*x+cosd(q)*y;
```

39.*a*

Command Window:

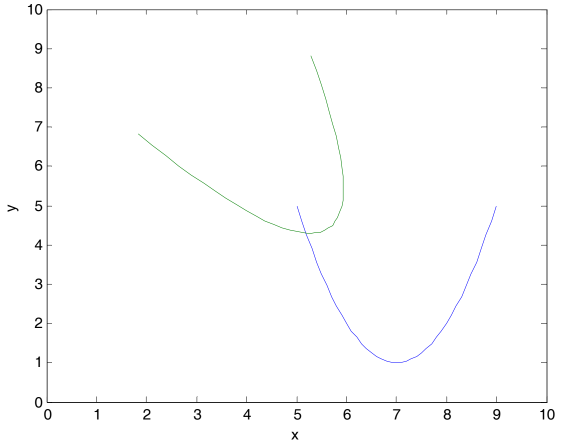
```
>> [xr,yr] = rotation(6.5,2.1,25)
xr =
    5.0035
yr =
    4.6503
```

39.*b*

Script file:

```
clear, clc
F=@ (x) (x-7).^2+1;
x=5:0.1:9;
y=F(x);
q=30;
[xr,yr] = rotation(x,y,q);
plot(x,y,xr,yr)
axis([0 10 0 10])
xlabel('x')
ylabel('y')
```

Figure:


10
9
8
7
6
5
4
3
2
1
0
y
0
1
2
3
4
5
6
7
8
9
10
x