MATLAB PROJECT 2

Please include this page in your Group file, as a front page. Type in the group number and the names of all members WHO PARTICIPATED in this project.

GROUP # _____26______

FIRST & LAST NAMES (UFID numbers are NOT required):

_1. James Luberisse
_2. Hao Lin

_3. Ryan Houston

_4._William Liu

_5. Seadn Madsen

_6. Anthony Khmarin

By including your names above, each of you had confirmed that you did the work and agree with the work submitted.

Exercise 1

```
type ele1
```

```
function E1=ele1(n,r,i,j)
    m = eye(n);
    replacement = m(j,:) + (r * m(i,:));
    m(j,:) =  replacement;

    E1 = m;
end
```

```
type ele2
```

```
function E2=ele2(n,i,j)
 m = eye(n);
 temp = m(j,:);
 m(j,:) = m(i,:);
 m(i,:) = temp;
 E2 = m;

end
```

```
type ele3
```

```
function E3=ele3(n,j,k)
 m = eye(n);
 m(j,:) = m(j,:) * k;

 E3 = m;
end
```

PART1

(a)

```
n=4; r=5; i=1; j=3; k=2;
I=eye(4)
```

```
I = 4×4
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

```
E1 = ele1(n,r,i,j);
disp(E1)
```

```
     1     0     0     0
     0     1     0     0
     5     0     1     0
     0     0     0     1
```

```
%  For E1 the row 3 is replaced by ( row 3 + row 1 scaled by 5)
E2 = ele2(n,i,j);
disp(E2)
```

```
     0     0     1     0
     0     1     0     0
     1     0     0     0
     0     0     0     1
```

```
% For E2, row 3 and row 1 are intechanged (or switched)
E3 = ele3(n,j,k);
disp(E3)
```

```
     1     0     0     0
     0     1     0     0
     0     0     2     0
     0     0     0     1
```

```
% For E3, row 3 is scaled by 2.
```

(b)

```
detI=det(I)
```

```
detI = 1
```

```
% the determinant was calculated the triangular way. Thus the result is 1.
detE1=det(E1)
```

```
detE1 = 1
```

```
% the detreminat was calcaluated by the cofactor method. Thus the result is
% 1.
detE2=det(E2)
```

```
detE2 = -1
```

```
% the determinant was calculated by interchnaging rows. which later
% produced a negative when calculating the determinant by the triangular
% method.
detE3=det(E3)
```

```
detE3 = 2
```

```
%the determinant was calcaluted by the traingular method. Which resulted in
%2.
```

(c)

```
invE1=inv(E1)
```

```
invE1 = 4×4
     1     0     0     0
     0     1     0     0
    -5     0     1     0
     0     0     0     1
```

```
% All the rows remained almost the same, but entry E1(3,1) was negated.
invE2=inv(E2)
```

```
invE2 = 4×4
     0     0     1     0
     0     1     0     0
     1     0     0     0
     0     0     0     1
```

```
% All the rows remained the same.
invE3=inv(E3)
```

```
invE3 = 4×4
    1.0000         0         0         0
         0    1.0000         0         0
         0         0    0.5000         0
         0         0         0    1.0000
```

```
% All the rows remained the same but the entry E3(3,3) was replaced by its
% reciprocal.
```

(d)

```
M=[1 1 1 1; 2 2 2 2; 3 3 3 3; 4 4 4 4]
```

```
M = 4×4
     1     1     1     1
     2     2     2     2
     3     3     3     3
     4     4     4     4
```

```
E1*M
```

```
ans = 4×4
     1     1     1     1
     2     2     2     2
     8     8     8     8
     4     4     4     4
```

```
% R3 = (R3 + 5*R1)
E2*M
```

```
ans = 4×4
     3     3     3     3
     2     2     2     2
     1     1     1     1
     4     4     4     4
```

```
% R3 and R1 are interchanged.
E3*M
```

```
ans = 4×4
     1     1     1     1
     2     2     2     2
     6     6     6     6
     4     4     4     4
```

```
% R3 is scaled by 2.
```

PART2

```
A = eye(6)
```

```
A = 6×6
     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
```

```
     0     0     0     0     0     1
```

```
% the matrix A is invertible since its determinant is 1 and not equal to 0.
invA = eye(6)
```

```
invA = 6×6
     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
```

```

A = ele1(6,3,2,5)*A;
invA = invA*inv(ele1(6,3,2,5));

A =ele2(6,2,3)*A;
invA =  invA*inv(ele2(6,2,3));

A = ele3(6,4,5)*A;
invA = invA*inv(ele3(6,4,5));

disp(A)
```

```
     1     0     0     0     0     0
     0     0     1     0     0     0
     0     1     0     0     0     0
     0     0     0     5     0     0
     0     3     0     0     1     0
     0     0     0     0     0     1
```

```

inv1=inv(A)
```

```
inv1 = 6×6
    1.0000         0         0         0         0         0
         0         0    1.0000         0         0         0
         0    1.0000         0         0         0         0
         0         0         0    0.2000         0         0
         0         0   -3.0000         0    1.0000         0
         0         0         0         0         0    1.0000
```

```
inv2 = invA
```

```
inv2 = 6×6
    1.0000         0         0         0         0         0
         0         0    1.0000         0         0         0
         0    1.0000         0         0         0         0
         0         0         0    0.2000         0         0
         0         0   -3.0000         0    1.0000         0
         0         0         0         0         0    1.0000
```

```


if inv1 == inv2
    disp('inverses match')
else
    disp('check code')
```

```
end
```

```
inverses match
```

```

```

Exercise 2

```
format compact
type rredef
```

```
function R = rredef(A)
[m,n]=size(A);
R=A;

column = 1;
row = 1;
R = sortrows(R, 1,'descend','ComparisonMethod','abs');
while row <= m & column <= n
    if ~any(R(row:end,column))
        column = column + 1;
    else
        R = closetozeroroundoff(R,7);
        R = [R(1:row, :); sortrows(R(row+1:end,:),column,'descend','ComparisonMethod','abs')];
        if R(row,column) ~= 0
            R(row,:) = R(row,:) / R(row,column);
            for current = row+1:m
                R(current,:) = R(current,:) - R(row,:)*R(current,column);
            end
        end
        row = row + 1;
        column = column + 1;
    end
end
R = closetozeroroundoff(R,7);
for row = m:-1:1
    if any(R(row,:))
        cV = find(R(row,:));
        column = cV(1);
        for current = row-1:-1:1
            R(current,:) = R(current,:) - R(row,:)*R(current,column);
        end
        R = closetozeroroundoff(R,7);
    end
end

r = rref(A);
if(closetozeroroundoff(R-r,7)==0)
    disp('the reduced echelon form of A is')
    R
else
    disp('Something is wrong!')
end
```

```
type closetozeroroundoff
```

```
function B=closetozeroroundoff(A,p)
A(abs(A)<10^-p)=0;
B=A;
end
```

```
%(a)
```

```
A=[2 1 1;1 2 3;1 1 1]
```

```
A = 3×3
     2     1     1
     1     2     3
     1     1     1
```

```
R=rredef(A);
```

```
the reduced echelon form of A is
R = 3×3
     1     0     0
     0     1     0
     0     0     1
```

```
%(b)
A=[zeros(3),randi(10,3,2)]
```

```
A = 3×5
     0     0     0     8     5
     0     0     0     5     4
     0     0     0     5     6
```

```
R=rredef(A);
```

```
the reduced echelon form of A is
R = 3×5
     0     0     0     1     0
     0     0     0     0     1
     0     0     0     0     0
```

```
%(c)
A=magic(4)
```

```
A = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
R=rredef(A);
```

```
the reduced echelon form of A is
R = 4×4
    1.0000         0         0    1.0000
         0    1.0000         0    3.0000
         0         0    1.0000   -3.0000
         0         0         0         0
```

```
%(d)
A=magic(5)
```

```
A = 5×5
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

```
R=rredef(A);
```

```
the reduced echelon form of A is
```

```
R = 5×5
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
```

```
%(e)
A=ones(3)
```

```
A = 3×3
     1     1     1
     1     1     1
     1     1     1
```

```
R=rredef(A);
```

```
the reduced echelon form of A is
R = 3×3
     1     1     1
     0     0     0
     0     0     0
```

```
%(f)
A=rand(3,4)
```

```
A = 3×4
    0.5108    0.6443    0.5328    0.8759
    0.8176    0.3786    0.3507    0.5502
    0.7948    0.8116    0.9390    0.6225
```

```
R=rredef(A);
```

```
the reduced echelon form of A is
R = 3×4
    1.0000         0         0    0.2076
         0    1.0000         0    2.7766
         0         0    1.0000   -1.9126
```

```
%(g)
A=randi(10,5,3);
A=[A,A(:,3)]
```

```
A = 5×4
     6     9     5     5
     3     2     4     4
     4     3    10    10
     5     2     5     5
     3     3     2     2
```

```
R=rredef(A);
```

```
the reduced echelon form of A is
R = 5×4
     1     0     0     0
     0     1     0     0
     0     0     1     1
     0     0     0     0
     0     0     0     0
```

Exercise 3

```
format compact
type homobasis_b
```

```
function [C,p] = homobasis_b(A,b)
format
[m,n]=size(A);
red_ech_form = rats(rref(A));
C=[];

[~,pivot_c] = rref(A);
S=1:n;
nonpivot_c = setdiff(S,pivot_c);
q=numel(nonpivot_c);
j=1:q;
fprintf('a free variable is x%i\n',nonpivot_c(j))
C=zeros(n,q);
Aref = rref(A);
colOfZero = 0;
for k = 1:n
    if Aref(:,k) == zeros(m,1)
        colOfZero = 1 + colOfZero;
    else
        break;
    end
end
C(1+colOfZero:m+colOfZero,j)= -Aref(1:m,nonpivot_c(j));
for i = 1:q
 C(nonpivot_c(i),i) = 1;
end
if isequal(rank(C),q) && isequal(closetozeroroundoff(A*C,5),zeros(m,q))
    fprintf('a basis for the solution set of the homogeneous system\n')
    fprintf('is formed by the columns of the matrix')
    C
else
    disp('Alert, bug detected')
    C = [];
end
[~,pivot_c] = rref([A, b]);

R = rref([A,b]);
p = zeros(size(C,1),1);

for t = 1:numel(pivot_c)
    temp_c = pivot_c(t);
    pivot_r = 0;
    for z = 1:size(R,1)
        if R(z,temp_c) == 1
            pivot_r = z;
        end
    end
    p(temp_c,1) = R(pivot_r,size(R,2));
end
disp('particular solution of the non-homogeneous system is the vector')
p
end
```

```
type nonhomogen
```

```
function x=nonhomogen(A,b)
format
```

```
[~,n]=size(A);
fprintf('reduced echelon form of [A b] is ')
R=rref([A,b])
x=[];
if rank(A) == rank(R)
    if isequal(rank(A), n)
        disp('The system has a unique solution')
        x = A\b
    else
        disp('There are infinitely many solutions')
        [C,p] = homobasis_b(A,b);
        syms Col(C), syms p
        fprintf('the general solution of the non-homogeneous system is\n')
        fprintf('the column space of the matrix C translated by the vector p')
        x = Col(C)+p
    end

else
    disp('The system is inconsistent')
end
```

```
%(a)
A=[1 -2 3], b=randi(10,1,1)
```

```
A = 1×3
     1    -2     3
b = 10
```

```
x=nonhomogen(A,b);
```

```
reduced echelon form of [A b] is
R = 1×4
     1    -2     3    10
There are infinitely many solutions
a free variable is x2
a free variable is x3
a basis for the solution set of the homogeneous system
is formed by the columns of the matrix
C = 3×2
     2    -3
     1     0
     0     1
particular solution of the non-homogeneous system is the vector
p = 3×1
    10
     0
     0
the general solution of the non-homogeneous system is
the column space of the matrix C translated by the vector p
```

x = $p + \mathrm{Col}(C)$

```
%(b)
A=magic(3), b=randi(10,3,1)
```

```
A = 3×3
     8     1     6
     3     5     7
     4     9     2
b = 3×1
    10
     5
```

```
     2
```

```
x=nonhomogen(A,b);
```

```
reduced echelon form of [A b] is
R = 3×4
    1.0000         0         0    0.8778
         0    1.0000         0   -0.2889
         0         0    1.0000    0.5444
The system has a unique solution
x = 3×1
    0.8778
   -0.2889
    0.5444
```

```
%(c)
A=magic(4), b=randi(10,4,1)
```

```
A = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
b = 4×1
     3
     5
     6
     3
```

```
x=nonhomogen(A,b);
```

```
reduced echelon form of [A b] is
R = 4×5
     1     0     0     1     0
     0     1     0     3     0
     0     0     1    -3     0
     0     0     0     0     1
The system is inconsistent
```

```
%(d)
B=[0 1 2 3;0 2 4 6]; A=[B; eye(4)], b=sum(A,2)
```

```
A = 6×4
     0     1     2     3
     0     2     4     6
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
b = 6×1
     6
    12
     1
     1
     1
     1
```

```
x=nonhomogen(A,b);
```

```
reduced echelon form of [A b] is
R = 6×5
     1     0     0     0     1
     0     1     0     0     1
```

```
     0     0     1     0     1
     0     0     0     1     1
     0     0     0     0     0
     0     0     0     0     0
The system has a unique solution
x = 4×1
    1.0000
    1.0000
    1.0000
    1.0000
```

```
%(e)
A=[0 1 0 2 0 3; 0 2 0 4 0 6; 0 4 0 8 0 6], b=ones(3,1)
```

```
A = 3×6
     0     1     0     2     0     3
     0     2     0     4     0     6
     0     4     0     8     0     6
b = 3×1
     1
     1
     1
```

```
x=nonhomogen(A,b);
```

```
reduced echelon form of [A b] is
R = 3×7
     0     1     0     2     0     0     0
     0     0     0     0     0     1     0
     0     0     0     0     0     0     1
The system is inconsistent
```

```
%(f)
A=[0 1 0 2 0 3; 0 2 0 4 0 6; 0 4 0 8 0 6], b=sum(A,2)
```

```
A = 3×6
     0     1     0     2     0     3
     0     2     0     4     0     6
     0     4     0     8     0     6
b = 3×1
     6
    12
    18
```

```
x=nonhomogen(A,b);
```

```
reduced echelon form of [A b] is
R = 3×7
     0     1     0     2     0     0     3
     0     0     0     0     0     1     1
     0     0     0     0     0     0     0
There are infinitely many solutions
a free variable is x1
a free variable is x3
a free variable is x4
a free variable is x5
a basis for the solution set of the homogeneous system
is formed by the columns of the matrix
C = 6×4
     1     0     0     0
     0     0    -2     0
     0     1     0     0
     0     0     1     0
```

```
     0     0     0     1
     0     0     0     0
particular solution of the non-homogeneous system is the vector
p = 6×1
     0
     3
     0
     0
     0
     1
the general solution of the non-homogeneous system is
the column space of the matrix C translated by the vector p
```

x = $p + \mathrm{Col}(C)$

```
%(g)
A=[0 0 1 2 3;0 0 2 4 5], b=A(:,end)
```

```
A = 2×5
     0     0     1     2     3
     0     0     2     4     5
b = 2×1
     3
     5
```

```
x=nonhomogen(A,b);
```

```
reduced echelon form of [A b] is
R = 2×6
     0     0     1     2     0     0
     0     0     0     0     1     1
There are infinitely many solutions
a free variable is x1
a free variable is x2
a free variable is x4
a basis for the solution set of the homogeneous system
is formed by the columns of the matrix
C = 5×3
     1     0     0
     0     1     0
     0     0    -2
     0     0     1
     0     0     0
particular solution of the non-homogeneous system is the vector
p = 5×1
     0
     0
     0
     0
     1
the general solution of the non-homogeneous system is
the column space of the matrix C translated by the vector p
```

x = $p + \mathrm{Col}(C)$

```
%(h)
A=[0 0 1 2 3;0 0 2 4 6], b=A(:,end)
```

```
A = 2×5
     0     0     1     2     3
     0     0     2     4     6
b = 2×1
     3
     6
```

```
x=nonhomogen(A,b);
```

```
reduced echelon form of [A b] is
R = 2×6
     0     0     1     2     3     3
     0     0     0     0     0     0
There are infinitely many solutions
a free variable is x1
a free variable is x2
a free variable is x4
a free variable is x5
a basis for the solution set of the homogeneous system
is formed by the columns of the matrix
C = 5×4
     1     0     0     0
     0     1     0     0
     0     0    -2    -3
     0     0     1     0
     0     0     0     1
particular solution of the non-homogeneous system is the vector
p = 5×1
     0
     0
     3
     0
     0
the general solution of the non-homogeneous system is
the column space of the matrix C translated by the vector p
```

x = $p + \mathrm{Col}(C)$

Exercise 4

```
type areavol.m
```

```
function D=areavol(A)
format
D=0;

isParallelogram = 0;
if size(A,2) == 2
    isParallelogram=1;
end

rank(A);
size(A);
if size(A) > rank(A)
    if isParallelogram == 1
        fprintf("Parallelogram cannot be built.\n");
    else
        fprintf("Parallelipiped cannot be built.\n");
    end
    D=0;
return;
end

D = abs(det(A));

if isParallelogram == 1
    fprintf('The area of the parallelogram is\n');
    D
else
    fprintf('The volume of the parallelepiped is\n');
```

```
    D
end
end
```

```
%(a)
A=eye(2)
```

```
A = 2×2
     1     0
     0     1
```

```
D=areavol(A);
```

```
The area of the parallelogram is
D = 1
```

```
%(b)
A=magic(3)
```

```
A = 3×3
     8     1     6
     3     5     7
     4     9     2
```

```
D=areavol(A);
```

```
The volume of the parallelepiped is
D = 360
```

```
%(c)
A=randi(10,2)
```

```
A = 2×2
     7     3
     8     2
```

```
D=areavol(A);
```

```
The area of the parallelogram is
D = 10
```

```
%(d)
A=fix(10*rand(3))
```

```
A = 3×3
     2     5     8
     3     0     0
     4     2     9
```

```
D=areavol(A);
```

```
The volume of the parallelepiped is
D = 87
```

```
%(e)
B=randi([-10,10],2,1); A = [B,3*B]
```

```
A = 2×2
     5    15
     0     0
```

```
D=areavol(A);
```

```
Parallelogram cannot be built.
```

```
%(f)
X=randi([-10,10],3,1);Y=randi([-10,10],3,1);A=[X,Y,X+Y]
```

```
A = 3×3
     2    10    12
    -6     1    -5
    -1     0    -1
```

```
D=areavol(A);
```

```
Parallelipiped cannot be built.
```

Exercise 5

```
type transf.m
```

```
function C=transf(A,E)
C=A*E;
x=C(1,:);y=C(2,:);
plot(x,y)
v=[-5 5 -5 5];
axis(v)
end
```

```
E=[0 1 1 0 0;0 0 1 1 0];
A=eye(2);
hold on
grid on
E=transf(A,E)
```

```
E = 2×5
     0     1     1     0     0
     0     0     1     1     0
```

```
VS=[1 0;3 1];
A=VS;
E=transf(A,E)
```

```
E = 2×5
     0     1     1     0     0
     0     3     4     1     0
```

```
RS=[0 1;1 0];
A=RS;
E=transf(A,E)
```

```
E = 2×5
     0     3     4     1     0
     0     1     1     0     0
```

```
RX=[1 0;0 -1];
A=RX;
E=transf(A,E)
```

```
E = 2×5
     0     3     4     1     0
     0    -1    -1     0     0
```

```
RA=[0 -1;-1 0];
A=RA;
E=transf(A,E)
```

```
E = 2×5
     0     1     1     0     0
     0    -3    -4    -1     0
```

```
RY=[-1 0;0 1];
A=RY;
E=transf(A,E)
```

```
E = 2×5
     0    -1    -1     0     0
     0    -3    -4    -1     0
```

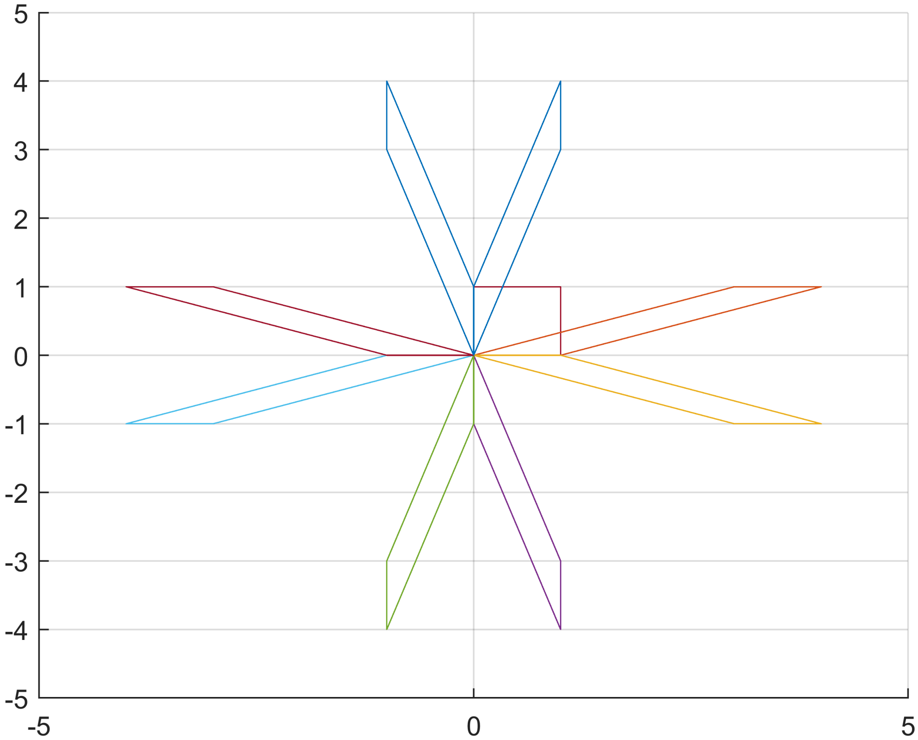
```
RS=[0 1;1 0];
A=RS;
E=transf(A,E)
```

```
E = 2×5
     0    -3    -4    -1     0
     0    -1    -1     0     0
```

```
RX=[1 0;0 -1];
A=RX;
E=transf(A,E)
```

```
E = 2×5
     0    -3    -4    -1     0
     0     1     1     0     0
```

```
RA=[0 -1;-1 0];
A=RA;
E=transf(A,E)
```




```
E = 2×5
     0    -1    -1     0     0
     0     3     4     1     0
```

Exercise 6

```
type closetozeroroundoff
```

```
function B=closetozeroroundoff(A,p)
A(abs(A)<10^-p)=0;
B=A;
end
```

```
type cofactor
```

```
function C=cofactor(a)
format
format compact
[~,n]=size(a);

 C=[];
   A=a;
 for i=1:n
    for j=1:n
        A(i,:)=[];
        A(:,j)=[];
        C(i,j)=(-1)^(i+j)*det(A);
        A=a;
```

```
    end
 end


disp('the matrix of cofactors of a is')
disp(C);
end
```

```
type determine
```

```
function D=determine(a,C)
D=[];

n=size(a,1);
% disp('a =');
% disp(a);
% disp(C);
% disp(rank(a));

% [m,n]=size(a);

 if(rank(a)~=size(a))
   disp('the determinant of the matrix a is')
  D =0;
  disp(D);
   return;
 end

E = zeros(n,2);
for i=1:n
    E(i,1)=sum(a(i,:).*C(i,:));
    E(i,2)= sum(a(:,i).*C(:,i));
    if(closetozeroroundoff(E(i,1)- E(i,2),7))
         disp('Something wrong with my code ');
         break;
    end
end
%  end
%   for i=1:n
%   D1(i)=sum(a(i,:).*C(i,:));
%  end
%   for j=1:n
%   D2(j)=sum(a(:,j).*C(:,j));
%   end
%  for k=1:n
%  if D1(k)~=D2(k)
%      disp('Something wrong with my code ');
%   break;
%  end
  if i==n
     D=E(1,1);
     disp('the determinant of the matrix is')
     disp(D);
  end
end
```

```
type inverse
```

```
function B=inverse(a,C,D)
B = [];
if(D == 0)
         disp('Matrix A is not invertible');
         return;
```

```
end
% if(size(D)==size(C))
% disp('B =');
% disp(B);
% disp(D);
 %T = transpose(C);
%  for i=1:size(C)
%      for j=1:size(C)
%        B(i,j)= (1/D)*T(i,j);
%      end
%  end
%  [m, n]=size(a);
%  if (rank(a)==m)
%  B=C'/D;
%  else
%  B=[];
%  end

B = (1 / D)*transpose(C);
% disp((1/D)*transpose(C));
F = inv(a);
disp(F);
B = closetozeroroundoff(B,7);
F = closetozeroroundoff(F,7);
% else
%     disp('What is wrong?');
%     return;
% end
if(closetozeroroundoff(B-F,7))
    disp('What is wrong?');
    disp(B);
    disp('');
    disp(F);
    return;
end
   disp('The inverse is calculated correctly and it is matrix');
   disp(B);

end
```

```
%a
 a=diag([1,2,3,4,5])
```

```
a = 5×5
     1     0     0     0     0
     0     2     0     0     0
     0     0     3     0     0
     0     0     0     4     0
     0     0     0     0     5
```

```
 C=cofactor(a);
```

```
the matrix of cofactors of a is
   120     0     0     0     0
     0    60     0     0     0
     0     0    40     0     0
     0     0     0    30     0
     0     0     0     0    24
```

```
 D=determine(a,C);
```

```
the determinant of the matrix is
   120
```

```
B=inverse(a,C,D);
```

```
    1.0000         0         0         0         0
         0    0.5000         0         0         0
         0         0    0.3333         0         0
         0         0         0    0.2500         0
         0         0         0         0    0.2000
The inverse is calculated correctly and it is matrix
    1.0000         0         0         0         0
         0    0.5000         0         0         0
         0         0    0.3333         0         0
         0         0         0    0.2500         0
         0         0         0         0    0.2000
```

```
%(b)
  a=ones(4)
```

```
a = 4×4
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

```
C=cofactor(a);
```

```
the matrix of cofactors of a is
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

```
D=determine(a,C);
```

```
the determinant of the matrix a is
     0
```

```
B=inverse(a,C,D);
```

```
Matrix A is not invertible
```

```
%(c)
a=magic(3)
```

```
a = 3×3
     8     1     6
     3     5     7
     4     9     2
```

```
C=cofactor(a);
```

```
the matrix of cofactors of a is
   -53    22     7
    52    -8   -68
   -23   -38    37
```

```
D=determine(a,C);
```

```
the determinant of the matrix is
```

```
  -360
```

```
B=inverse(a,C,D);
```

```
    0.1472   -0.1444    0.0639
   -0.0611    0.0222    0.1056
   -0.0194    0.1889   -0.1028
The inverse is calculated correctly and it is matrix
    0.1472   -0.1444    0.0639
   -0.0611    0.0222    0.1056
   -0.0194    0.1889   -0.1028
```

```
%(d)
a=magic(4)
```

```
a = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
C=cofactor(a);
```

```
the matrix of cofactors of a is
   1.0e+03 *
   -0.1360   -0.4080    0.4080    0.1360
   -0.4080   -1.2240    1.2240    0.4080
    0.4080    1.2240   -1.2240   -0.4080
    0.1360    0.4080   -0.4080   -0.1360
```

```
D=determine(a,C);
```

```
the determinant of the matrix a is
     0
```

```
B=inverse(a,C,D);
```

```
Matrix A is not invertible
```

```
%(e)
a=hilb(4)
```

```
a = 4×4
    1.0000    0.5000    0.3333    0.2500
    0.5000    0.3333    0.2500    0.2000
    0.3333    0.2500    0.2000    0.1667
    0.2500    0.2000    0.1667    0.1429
```

```
C=cofactor(a);
```

```
the matrix of cofactors of a is
    0.0000   -0.0000    0.0000   -0.0000
   -0.0000    0.0002   -0.0004    0.0003
    0.0000   -0.0004    0.0011   -0.0007
   -0.0000    0.0003   -0.0007    0.0005
```

```
D=determine(a,C);
```

```
the determinant of the matrix is
   1.6534e-07
```

```
B=inverse(a,C,D);
```

```
   1.0e+03 *
    0.0160   -0.1200    0.2400   -0.1400
   -0.1200    1.2000   -2.7000    1.6800
    0.2400   -2.7000    6.4800   -4.2000
   -0.1400    1.6800   -4.2000    2.8000
The inverse is calculated correctly and it is matrix
   1.0e+03 *
    0.0160   -0.1200    0.2400   -0.1400
   -0.1200    1.2000   -2.7000    1.6800
    0.2400   -2.7000    6.4800   -4.2000
   -0.1400    1.6800   -4.2000    2.8000
```

Exercise 7

```
type production
```

```
function x = production(C,d)
n = size(C,2);
x = [];

%Check for valid input
%First check C has all positive numbers
invalidInput = false;
for i = 1:n
    for j = 1:n
        if C(i,j) < 0
            disp('C has a negative number: invalid input')
            invalidInput = true;
        end
    end
end
%Second check d has all positive numbers
for i = 1:n
    if d(i) < 0
        disp('D has a negative number: invalid input')
        invalidInput = true;
    end
end
%Third check each column sum of C is less than 1
Sum = sum(C);
for i = 1:n
    if Sum(i) > 1
        disp('column sum in C is greater than 1: invalid input')
        invalidInput = true;
    end
end
if invalidInput
    return
end

%Input is Valid: Output the production vector (I-C)x = d
%calculate I - C
IdentityMatrix = eye(n);
x = (IdentityMatrix - C) \ d;
%Verify x is economically feasible (positive numbers)
if all(x>=0)
    disp('the unique production vector is')
```

```
    x
else
    disp('check the code!')
    x=[];
    return
end

%Find x using equation from page 139 - find number of iterations
x0 = d;
x1 = 0;
xPrevious =d;
k=0;
while all(closetozeroroundoff(single(x-x1),1) ~= 0)
    x1 = C*xPrevious + d;
    xPrevious = x1;
    k = k+1;
end

disp('the production vector calculated by recurrence relation is')
x1
fprintf('the number of iterations to match the output x is %i\n',k)
```

```
% (a)
C = [0.5 0.4 0.2; 0.2 0.3 0.1; 0.1 0.1 0.3]
```

```
C = 3×3
    0.5000    0.4000    0.2000
    0.2000    0.3000    0.1000
    0.1000    0.1000    0.3000
```

```
D = [50; 30; 20]
```

```
D = 3×1
    50
    30
    20
```

```
x = production(C,D);
```

```
the unique production vector is
x = 3×1
  225.9259
  118.5185
   77.7778
the production vector calculated by recurrence relation is
x1 = 3×1
  225.6447
  118.3779
   77.6870
the number of iterations to match the output x is 24
```

```
% (b)
C = importdata('consumption.csv')
```

```
C = 7×7
    0.1588    0.0064    0.0025    0.0304    0.0014    0.0083    0.1504
    0.0057    0.2645    0.0436    0.0099    0.0083    0.0201    0.3413
    0.0264    0.1506    0.3557    0.0139    0.0142    0.0070    0.0236
    0.3299    0.0565    0.0495    0.3636    0.0204    0.0483    0.0649
```

```
    0.0089    0.0081    0.0333    0.0295    0.3412    0.0237    0.0020
    0.1190    0.0901    0.0996    0.1260    0.1722    0.2368    0.3369
    0.0063    0.0126    0.0196    0.0098    0.0064    0.0132    0.0012
```

```
d = importdata('demand.csv')
```

```
d = 7×1
       74000
       56000
       10500
       25000
       17500
      196000
        5000
```

```
x = production(C,d);
```

```
the unique production vector is
x = 7×1
    0.9942
    0.9770
    0.5122
    1.3149
    0.4948
    3.2951
    0.1383
the production vector calculated by recurrence relation is
x1 = 7×1
    0.9942
    0.9770
    0.5122
    1.3149
    0.4948
    3.2951
    0.1383
the number of iterations to match the output x is 18
```

```
% (c)
C = importdata('consumption.csv')
```

```
C = 7×7
    0.1588    0.0064    0.0025    0.0304    0.0014    0.0083    0.1504
    0.0057    0.2645    0.0436    0.0099    0.0083    0.0201    0.3413
    0.0264    0.1506    0.3557    0.0139    0.0142    0.0070    0.0236
    0.3299    0.0565    0.0495    0.3636    0.0204    0.0483    0.0649
    0.0089    0.0081    0.0333    0.0295    0.3412    0.0237    0.0020
    0.1190    0.0901    0.0996    0.1260    0.1722    0.2368    0.3369
    0.0063    0.0126    0.0196    0.0098    0.0064    0.0132    0.0012
```

```
d = importdata('demand_1.csv')
```

```
d = 7×1
       99640
       75548
       14444
       33501
       23527
      263985
        6526
```

```
x = production(C,d);
```

```
the unique production vector is
```

```
x = 7×1
    1.3383
    1.3168
    0.6946
    1.7680
    0.6659
    4.4372
    0.1843
the production vector calculated by recurrence relation is
x1 = 7×1
    1.3383
    1.3168
    0.6946
    1.7680
    0.6658
    4.4372
    0.1843
the number of iterations to match the output x is 19
```

```
% (d)
C = importdata('consumption_1.csv')
```

```
C = 7×7
    1.1588    0.0064    0.0025    0.0304    0.0014    0.0083    0.1504
    0.0057    0.2645    0.0436    0.0099    0.0083    0.0201    0.3413
    0.0264    0.1506    0.3557    0.0139    0.0142    0.0070    0.0236
    0.3299    0.0565    0.0495    0.3636    0.0204    0.0483    0.0649
    0.0089    0.0081    0.0333    0.0295    0.3412    0.0237    0.0020
    0.1190    0.0901    0.0996    0.1260    0.1722    0.2368    0.3369
    0.0063    0.0126    0.0196    0.0098    0.0064    0.0132    0.0012
```

```
d = importdata('demand_1.csv')
```

```
d = 7×1
       99640
       75548
       14444
       33501
       23527
      263985
        6526
```

```
x = production(C,d);
```

```
column sum in C is greater than 1: invalid input
```

```
% (e)
C = importdata('consumption_1.csv')
```

```
C = 7×7
    1.1588    0.0064    0.0025    0.0304    0.0014    0.0083    0.1504
    0.0057    0.2645    0.0436    0.0099    0.0083    0.0201    0.3413
    0.0264    0.1506    0.3557    0.0139    0.0142    0.0070    0.0236
    0.3299    0.0565    0.0495    0.3636    0.0204    0.0483    0.0649
    0.0089    0.0081    0.0333    0.0295    0.3412    0.0237    0.0020
    0.1190    0.0901    0.0996    0.1260    0.1722    0.2368    0.3369
    0.0063    0.0126    0.0196    0.0098    0.0064    0.0132    0.0012
```

```
d = importdata('demand_2.csv')
```

```
d = 7×1
       99640
       75548
```

```
       14444
       33501
       23527
      263985
       -6526
```

```
x = production(C,d);
```

```
D has a negative number: invalid input
column sum in C is greater than 1: invalid input
```