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 # 4

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

_1._Edwin Salcedo

_2._Anisha Paul

_3._Brandon Tran

_4._John Dinh

_5._Ananya Kakaveti

_6._Yaaseen Mohammad

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

Exercise 1

```
type rredf
```

```
function R=rredf(A)
format
[m,n]=size(A);
rankA=rank(A);
A=sym(A);
R=A;

% Indexes for loop
i=1;
j=1;

% Forward Phase
for k=1:m
    while i<=m && j<=n
        % Getting largest by absolute value entry
        [x,y]=max(abs(R(i:m,j)));
        y=y+i-1;

        if closetozeroroundoff(R(i:m,j),7)==0
            closetozeroroundoff(R(i:m,j),7);
            j=j+1;
        else
            % Row interchanging operation
            R([i y],j:n)=R([y i],j:n);
            R(i,j:n)=R(i,j:n)./R(i,j);

            % Backward Phase
            for k=[1:i-1 i+1:m]
                R(k,j:n)=R(k,j:n)-R(k,j).*R(i,j:n);
            end
            i=i+1;
            j=j+1;
        end
    end
end

disp('the constructed matrix R is')
disp(double(R))

% Verifying that R is the reduced echelon form of A
rf=rref(A);

if closetozeroroundoff(R-rf,7)==0
    disp('R is the reduced echelon form of A')
    R=double(R);
else
    disp('Something went wrong!')
    R=[]
end
end
```

```
type closetozeroroundoff
```

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

```


%(a)
```

```
A=[2 0 1 3]
```

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

```
R=rredf(A);
```

```
the constructed matrix R is
    1.0000         0    0.5000    1.5000

R is the reduced echelon form of A
```

```
%(b)
A=[2 4 1;1 2 3;1 2 1]
```

```
A = 3×3
     2     4     1
     1     2     3
     1     2     1
```

```
R=rredf(A);
```

```
the constructed matrix R is
     1     2     0
     0     0     1
     0     0     0

R is the reduced echelon form of A
```

```
%(c)
A=[zeros(4),magic(4)]
```

```
A = 4×8
     0     0     0     0    16     2     3    13
     0     0     0     0     5    11    10     8
     0     0     0     0     9     7     6    12
     0     0     0     0     4    14    15     1
```

```
R=rredf(A);
```

```
the constructed matrix R is
     0     0     0     0     1     0     0     1
     0     0     0     0     0     1     0     3
     0     0     0     0     0     0     1    -3
     0     0     0     0     0     0     0     0

R is the reduced echelon form of A
```

```
%(d)
A=pascal(3)
```

```
A = 3×3
     1     1     1
     1     2     3
     1     3     6
```

```
R=rredf(A);
```

```
the constructed matrix R is
     1     0     0
     0     1     0
     0     0     1
```

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

```
%(e)
A=ones(3,6); A(:,1:2:5)=magic(3)
```

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

```
R=rredf(A);
```

```
the constructed matrix R is
     1     0     0     0    -1     0
     0     1     0     1    15     1
     0     0     1     0    -1     0

R is the reduced echelon form of A
```

```
%(f)
A=zeros(3,6); A(:,2:2:6)=magic(3)
```

```
A = 3×6
     0     8     0     1     0     6
     0     3     0     5     0     7
     0     4     0     9     0     2
```

```
R=rredf(A);
```

```
the constructed matrix R is
     0     1     0     0     0     0
     0     0     0     1     0     0
     0     0     0     0     0     1

R is the reduced echelon form of A
```

```
%(g)
A=[magic(4);hilb(4)]
```

```
A = 8×4
   16.0000    2.0000    3.0000   13.0000
    5.0000   11.0000   10.0000    8.0000
    9.0000    7.0000    6.0000   12.0000
    4.0000   14.0000   15.0000    1.0000
    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
```

```
R=rredf(A);
```

```
the constructed matrix R is
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

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

```
%(h)
A=[magic(4),hilb(4)]
```

```
A = 4×8
   16.0000    2.0000    3.0000   13.0000    1.0000    0.5000    0.3333    0.2500
    5.0000   11.0000   10.0000    8.0000    0.5000    0.3333    0.2500    0.2000
    9.0000    7.0000    6.0000   12.0000    0.3333    0.2500    0.2000    0.1667
    4.0000   14.0000   15.0000    1.0000    0.2500    0.2000    0.1667    0.1429
```

```
R=rredf(A);
```

```
the constructed matrix R is
    1.0000         0         0    1.0000         0    0.0041    0.0052    0.0054
         0    1.0000         0    3.0000         0    0.0262    0.0253    0.0222
         0         0    1.0000   -3.0000         0   -0.0196   -0.0182   -0.0154
         0         0         0         0    1.0000    0.4400    0.2533    0.1657

R is the reduced echelon form of A
```

```
%(i)
A=randi(10,5,3);A=[A, sum(A,2)]
```

```
A = 5×4
     7     2     2    11
     7     3    10    20
     9     9     8    26
     9     1     6    16
     6     5     5    16
```

```
R=rredf(A);
```

```
the constructed matrix R is
     1     0     0     1
     0     1     0     1
     0     0     1     1
     0     0     0     0
     0     0     0     0

R is the reduced echelon form of A
```

Exercise 2

```
format
A=[0 2 -4 0; 2 2 -4 4; 1 1 2 -2]
```

```
A = 3×4
     0     2    -4     0
     2     2    -4     4
     1     1     2    -2
```

```
R=A;
R=ele2(3,1,3)*R
```

```
R = 3×4
     1     1     2    -2
     2     2    -4     4
     0     2    -4     0
```

```
R=ele1(3,1,2,-2)*R
```

```
R = 3×4
     1     1     2    -2
     0     0    -8     8
     0     2    -4     0
```

```
R=ele2(3,2,3)*R
```

```
R = 3×4
     1     1     2    -2
     0     2    -4     0
     0     0    -8     8
```

```
R=ele1(3,2,1,-0.5)*R
```

```
R = 3×4
     1     0     4    -2
     0     2    -4     0
     0     0    -8     8
```

```
R=ele1(3,3,2,-0.5)*R
```

```
R = 3×4
     1     0     4    -2
     0     2     0    -4
     0     0    -8     8
```

```
R=ele1(3,3,1,0.5)*R
```

```
R = 3×4
     1     0     0     2
     0     2     0    -4
     0     0    -8     8
```

```
R=ele3(3,3,-1/8)*R
```

```
R = 3×4
     1     0     0     2
     0     2     0    -4
     0     0     1    -1
```

```
R=ele3(3,2,1/2)*R
```

```
R = 3×4
     1     0     0     2
     0     1     0    -2
     0     0     1    -1
```

```
rf=rref(A);
if isequal(R,rf)
    disp('the reduced echelon form of A is')
    disp(R)

else
    disp('I need to check my coding!')
    R=[]
end
```

```
the reduced echelon form of A is
     1     0     0     2
     0     1     0    -2
     0     0     1    -1
```

```
R=ele3(3,2,2)*R;
R=ele3(3,3,-8)*R;
R=ele1(3,3,1,-0.5)*R;
R=ele1(3,3,2,0.5)*R;
R=ele1(3,2,1,0.5)*R;
R=ele2(3,2,3)*R;
R=ele1(3,1,2,2)*R;
R=ele2(3,1,3)*R
```

```
R = 3×4
     0     2    -4     0
     2     2    -4     4
     1     1     2    -2
```

```
if isequal(A,R)
    disp('the row reduction operations reversed correctly')
else
    disp('the inversion did not work! Hmm...?')
end
```

```
the row reduction operations reversed correctly
```

```
type redef.m
```

```
function R=redef(A)
format
[m,n]=size(A);
rankA=rank(A);
A=sym(A);
R=A;
% Indexes for while loop to not exceed number of rows and columns
i=1; % Row index
j=1; % Column index
% Forward Phase
% m = total rows, n = total columns
% For each row of matrix R
for k=1:m
 % While we don't exceed number of rows and columns
     while i<=m && j<=n
     % Getting largest by absolute value column entry to be a pivot
     [x,y]=max(abs(R(i:m,j)));
     y=y+i-1;

         % If that column entry is in the pivot position
         if closetozeroroundoff(R(i:m,j),7)==0
         closetozeroroundoff(R(i:m,j),7);
         j=j+1;
         % If that column entry is not in the pivot position
         else
         % Using row interchanging operation to move it into the pivot position

          %R([i y],j:n)=R([y i],j:n); (Original)
%         Complete ele2
          R=ele2(m,y,i)*R;


           %Dividing matrix R's pivots

           %R(i,j:n)=R(i,j:n)./R(i,j); (Original)
%          Complete ele3
           R=ele3(m,i,1/R(i,j))*R;

         % Backward Phase
         % Rows are subtracted by pivot row's scalar multiples
         for k=[1:i-1 i+1:m]
%              R(k,j:n)=R(k,j:n)-R(k,j).*R(i,j:n); (Original)
%           Complete ele1
            R=ele1(m,i,k,-R(k,j))*R;
         end
         % Incrementing the while loop
         i=i+1;
         j=j+1;
         end
     end
end
disp('the constructed matrix R is')
disp(double(R))
% Verifying that R is the reduced echelon form of A
rf=rref(A);
if closetozeroroundoff(R-rf,7)==0
     disp('R is the reduced echelon form of A')
     R=double(R);
else
     disp('Something went wrong!')
     R=[]
end
end
```

```
%test
A=[0 2 -4 0; 2 2 -4 4; 1 1 2 -2]
```

```
A = 3×4
     0     2    -4     0
     2     2    -4     4
     1     1     2    -2
```

```
R=redef(A);
```

```
the constructed matrix R is
     1     0     0     2
     0     1     0    -2
     0     0     1    -1

R is the reduced echelon form of A
```

```
%(a)
A=[2 0 1 3]
```

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

```
R=redef(A);
```

```
the constructed matrix R is
    1.0000         0    0.5000    1.5000

R is the reduced echelon form of A
```

```
%(b)
A=[2 4 1;1 2 3;1 2 1]
```

```
A = 3×3
     2     4     1
     1     2     3
     1     2     1
```

```
R=redef(A);
```

```
the constructed matrix R is
     1     2     0
     0     0     1
     0     0     0

R is the reduced echelon form of A
```

```
%(c)
A=[zeros(4),magic(4)]
```

```
A = 4×8
     0     0     0     0    16     2     3    13
     0     0     0     0     5    11    10     8
     0     0     0     0     9     7     6    12
     0     0     0     0     4    14    15     1
```

```
R=redef(A);
```

```
the constructed matrix R is
     0     0     0     0     1     0     0     1
     0     0     0     0     0     1     0     3
     0     0     0     0     0     0     1    -3
     0     0     0     0     0     0     0     0

R is the reduced echelon form of A
```

```
%(d)
A=pascal(3)
```

```
A = 3×3
     1     1     1
     1     2     3
     1     3     6
```

```
R=redef(A);
```

```
the constructed matrix R is
     1     0     0
     0     1     0
     0     0     1

R is the reduced echelon form of A
```

```
%(e)
A=ones(3,6); A(:,1:2:5)=magic(3)
```

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

```
R=redef(A);
```

```
the constructed matrix R is
     1     0     0     0    -1     0
     0     1     0     1    15     1
     0     0     1     0    -1     0

R is the reduced echelon form of A
```

```
%(f)
A=zeros(3,6); A(:,2:2:6)=magic(3)
```

```
A = 3×6
     0     8     0     1     0     6
     0     3     0     5     0     7
     0     4     0     9     0     2
```

```
R=redef(A);
```

```
the constructed matrix R is
     0     1     0     0     0     0
     0     0     0     1     0     0
     0     0     0     0     0     1

R is the reduced echelon form of A
```

```
%(g)
A=[magic(4);hilb(4)]
```

```
A = 8×4
   16.0000    2.0000    3.0000   13.0000
    5.0000   11.0000   10.0000    8.0000
    9.0000    7.0000    6.0000   12.0000
    4.0000   14.0000   15.0000    1.0000
    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
```

```
R=redef(A);
```

```
the constructed matrix R is
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0

R is the reduced echelon form of A
```

```
%(h)
A=[magic(4),hilb(4)]
```

```
A = 4×8
   16.0000    2.0000    3.0000   13.0000    1.0000    0.5000    0.3333    0.2500
    5.0000   11.0000   10.0000    8.0000    0.5000    0.3333    0.2500    0.2000
    9.0000    7.0000    6.0000   12.0000    0.3333    0.2500    0.2000    0.1667
    4.0000   14.0000   15.0000    1.0000    0.2500    0.2000    0.1667    0.1429
```

```
R=redef(A);
```

```
the constructed matrix R is
    1.0000         0         0    1.0000         0    0.0041    0.0052    0.0054
         0    1.0000         0    3.0000         0    0.0262    0.0253    0.0222
         0         0    1.0000   -3.0000         0   -0.0196   -0.0182   -0.0154
         0         0         0         0    1.0000    0.4400    0.2533    0.1657

R is the reduced echelon form of A
```

```
%(i)
A=randi(10,5,3);A=[A, sum(A,2)]
```

```
A = 5×4
     4     7    10    21
     6     4     9    19
     6     9     6    21
     9     6     7    22
     8     4     6    18
```

```
R=redef(A);
```

```
the constructed matrix R is
     1     0     0     1
     0     1     0     1
     0     0     1     1
     0     0     0     0
     0     0     0     0

R is the reduced echelon form of A
```

Exercise 3

```
type nonhomogen.m
```

```
function x=nonhomogen(A,b)
format
[~,n]=size(A);
x=[];

if (rank([A b]) ~= rank(A))
    disp('The system is inconsistent')
    return
else
    if (rank(A)==size(A,2))
        disp('The system has a unique solution')
        x=A\b
        return
    else
        disp('There are infinitely many solutions')
    end

end

C = homobasis(A)
if isempty(C)
    p=[]
    return
end

[B,pivot]=rref([A b]);
p=zeros(n,1);
p(pivot,1)=B(1:rank(A),end);

if (closetozeroroundoff(A*p-b,7) == 0)
    disp('particular solution of non-homogeneous system is vector')
    p
else
    disp('Check the code!')
    p=[]
    return
end

syms Col(C), syms p
fprintf(['the general solution of non-homogeneous system Ax=b is\n' ...
 'the Col(C) translated by the vector p:'])
x=Col(C)+p



end
```

```
type homobasis.m
```

```
function C = homobasis(A)
format
[~,n]=size(A);
R=rref(A);
rankA=rank(A);
if rankA==n
    disp('the homogeneous system has only the trivial solution')
    C=zeros(n,1);
else
    disp('the homogeneous system has non-trivial solutions')
    [~,pivot]=rref(A);
    nonpivot=setdiff(1:n,pivot);
    q=numel(nonpivot);
    j=1:q;
    fprintf('a free variable is x%i\n',nonpivot(j))
    C=zeros(n,q);
    R=R(1:rankA,nonpivot);
    C(pivot,:)=-R;
    C(nonpivot,:)=eye(q);

    if rank(C)== size(C,2) && ~any(closetozeroroundoff(A*C,5),'all')
        disp('columns of C form a basis for solution set of the homogeneous system')
    else
        C=[];
    end
end
```

```
type closetozeroroundoff.m
```

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

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

```
A = 1×3
     1     2    -3
b = 9
```

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

```
There are infinitely many solutions
the homogeneous system has non-trivial solutions
a free variable is x2
a free variable is x3
columns of C form a basis for solution set of the homogeneous system
C = 3×2
    -2     3
     1     0
     0     1
particular solution of non-homogeneous system is vector
p = 3×1
     9
     0
     0
the general solution of non-homogeneous system Ax=b is
the Col(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
     7
     8
```

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

```
The system has a unique solution
x = 3×1
    0.9722
    0.3889
    0.3056
```

```
%(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
     8
     4
     7
     2
```

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

```
The system is inconsistent
```

```
%(d)
A=magic(4), b=ones(4,1)
```

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

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

```
There are infinitely many solutions
the homogeneous system has non-trivial solutions
a free variable is x4
columns of C form a basis for solution set of the homogeneous system
C = 4×1
    -1
    -3
     3
     1
particular solution of non-homogeneous system is vector
p = 4×1
    0.0588
    0.1176
   -0.0588
         0
the general solution of non-homogeneous system Ax=b is
the Col(C) translated by the vector p:
```

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

```
%(e)
```

```
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);
```

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

```
%(f)
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);
```

```
The system is inconsistent
```

```
%(g)
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);
```

```
There are infinitely many solutions
the homogeneous system has non-trivial solutions
a free variable is x1
a free variable is x3
a free variable is x4
```

```
a free variable is x5
columns of C form a basis for solution set of the homogeneous system
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 non-homogeneous system is vector
p = 6×1
     0
     3
     0
     0
     0
     1
the general solution of non-homogeneous system Ax=b is
the Col(C) translated by the vector p:
```
x = $p+\mathrm{Col}(C)$

```
%(h)
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);
```

```
There are infinitely many solutions
the homogeneous system has non-trivial solutions
a free variable is x1
a free variable is x2
a free variable is x4
columns of C form a basis for solution set of the homogeneous system
C = 5×3
     1     0     0
     0     1     0
     0     0    -2
     0     0     1
     0     0     0
particular solution of non-homogeneous system is vector
p = 5×1
     0
     0
     0
     0
     1
the general solution of non-homogeneous system Ax=b is
the Col(C) translated by the vector p:
```
x = $p+\mathrm{Col}(C)$

```
%(i)
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);
```

```
There are infinitely many solutions
the homogeneous system has non-trivial solutions
a free variable is x1
a free variable is x2
a free variable is x4
a free variable is x5
columns of C form a basis for solution set of the homogeneous system
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 non-homogeneous system is vector
p = 5×1
     0
     0
     3
     0
     0
the general solution of non-homogeneous system Ax=b is
the Col(C) translated by the vector p:
```

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

Area, Volume, and Graphics in MATLAB

Exercise 4

```
type areavol.m
```

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

if size(A) > rank(A)
    if size(A,2) == 2
        disp('Parallelogram cannot be built because vectors are in R^2 and are linearly dependent.')
        D
    else
        if det(A)~=0
            disp('parallelepiped in R^3 cannot be built but parallelogram can')
            [~,pivot]=rref(A);
            D=norm(cross(A(:,(pivot(1,1))),A(:,(pivot(1,2)))));
            fprintf('area of parallelogram built on pivot columns of A is %i\n',D)
        else
            disp('either parallelogram or parallelepiped in R^3 cannot be built')
        end
    end
else
    if size(A,2) == 2
        D = (abs(det(A)));
        for i=1:(size(A,2))
            A(3,i)=0;
        end
        S = norm(cross(A(:,1),A(:,2)));
        D = round(D,12);
        S = round(S,12);
        if closetozeroroundoff(D,7) == closetozeroroundoff(S,7)
            fprintf('The area of the parallelogram is %i\n',D)
        else
            disp('What can go wrong?!')
            D=[]
        end
    else
        D = (abs(det(A)));
        V = abs(dot(A(:,1),cross(A(:,2),A(:,3))));
        D = round(D,12);
        V = round(V,12);
        if closetozeroroundoff(D,7) == closetozeroroundoff(V,7)
            fprintf('The volume of the parallelepiped is %i\n',D)
        else
            disp('Something did go wrong!')
            D=[]
        end
    end
end
```

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

```
A = 2×2
     1    10
     9     8
```

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

```
The area of the parallelogram is 82
```

```
%(b)
A=hilb(2)
```

```
A = 2×2
    1.0000    0.5000
    0.5000    0.3333
```

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

```
The area of the parallelogram is 8.333333e-02
```

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

```
A = 3×3
     0     7     6
     3     2     4
     7     2     6
```

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

```
The volume of the parallelepiped is 22
```

```
%(d)
A=pascal(2)
```

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

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

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

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

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

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

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

```
%(f)
A=hilb(3)
```

```
A = 3×3
    1.0000    0.5000    0.3333
    0.5000    0.3333    0.2500
    0.3333    0.2500    0.2000
```

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

```
The volume of the parallelepiped is 4.629630e-04
```

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

```
A = 2×2
    -6   -18
    -7   -21
```

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

```
Parallelogram cannot be built because vectors are in R^2 and are linearly dependent.
D = 0
```

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

```
A = 3×3
     7    -8    15
     6    -7    13
     9    -8    17
```

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

```
parallelepiped in R^3 cannot be built but parallelogram can
area of parallelogram built on pivot columns of A is 2.195450e+01
```

```
%(i)
A=[1 2 3;2 4 6;3 6 9]
```

```
A = 3×3
     1     2     3
     2     4     6
     3     6     9
```

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

```
either parallelogram or parallelepiped in R^3 cannot be built
D = 0
```

Exercise 5

Part I (Shear & Reflections)

```
RX = [1 0;0 -1]
```

```
RX = 2×2
     1     0
     0    -1
```

```
RY = [-1 0;0 1]
```

```
RY = 2×2
    -1     0
     0     1
```

```
VS = [1 0;3 1]
```

```
VS = 2×2
     1     0
     3     1
```

```
HS = [1 3; 0 1]
```

```
HS = 2×2
     1     3
     0     1
```

```
RS = [0 1; 1 0]
```

```
RS = 2×2
     0     1
     1     0
```

```
RA = [0 -1; -1 0]
```

```
RA = 2×2
     0    -1
    -1     0
```

```
type transf.m
```

```
function C=transf(A,E) % takes in two matrices
C=A*E;                 % Matrix E is transformed by matrix A
x=C(1,:);y=C(2,:);     % x = column 1 of C and y = column 2 of C
plot(x,y)              % connects the vectors of C to make a shape
v=[-5 5 -5 5];         % defines the boundaries for the 2-D graph
axis(v)                % sets specified limits of graph using vector v
end
```

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

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

```
hold on
grid on
E=[0 1 1 0 0; 0 0 1 1 0]
```

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

```
A = RX*HS;
E=transf(A,E)
```

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

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

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

```
E=[0 1 1 0 0;0 0 1 1 0];
A = RY*RX*VS;
E=transf(A,E)
```

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

```
E=[0 1 1 0 0;0 0 1 1 0];
A = RY*RX*HS;
E=transf(A,E)
```

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

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

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

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

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

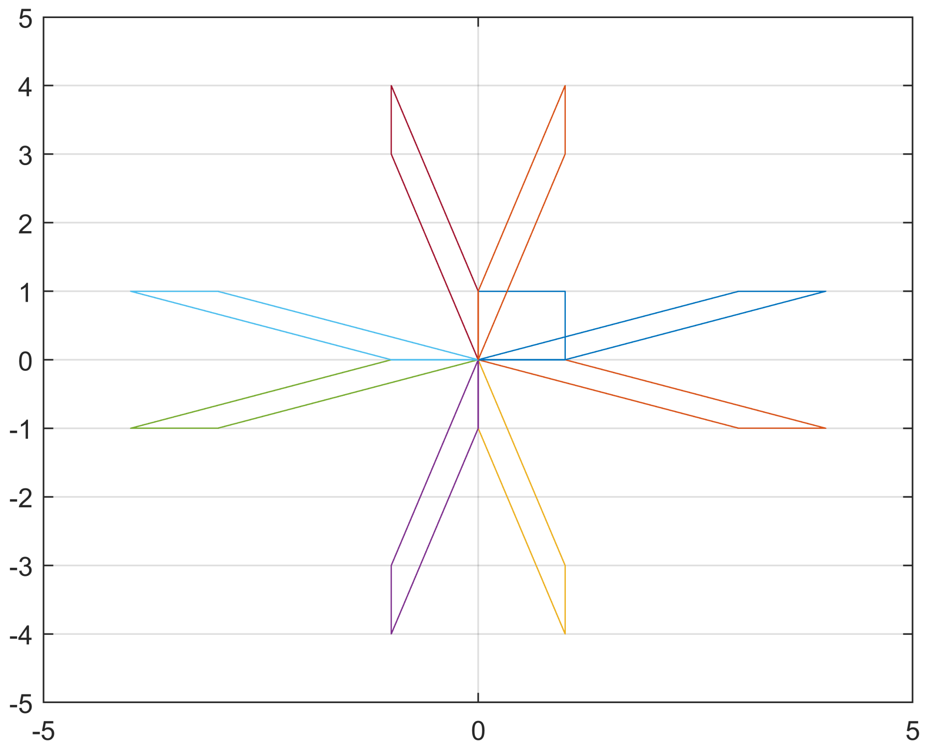
```
E=[0 1 1 0 0;0 0 1 1 0];
A = HS;
E=transf(A,E)
```

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

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

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

```
hold off
```



Part II (Shear & Rotation)

```
type givens.m
```

```
function G=givens(m,i,j,theta)
format
G=[];
% Givens works under the conditions (1 <= i < j<= m) and (m >= 2)
if (i >= 1)& (i < j) & (m >=2) & (j <= m)
    G = eye(m)
    c = cos(theta);
    s = sin(theta);
    G(i,i)=c; G(i,j)=-s; G(j,i)=s; G(j,j)=c;
    disp('the Givens rotation matrix G is')
    disp(G)
else
    disp('Givens rotation matrix cannot be constructed')
    return %terminates function
end
```

```
type rotation.m
```

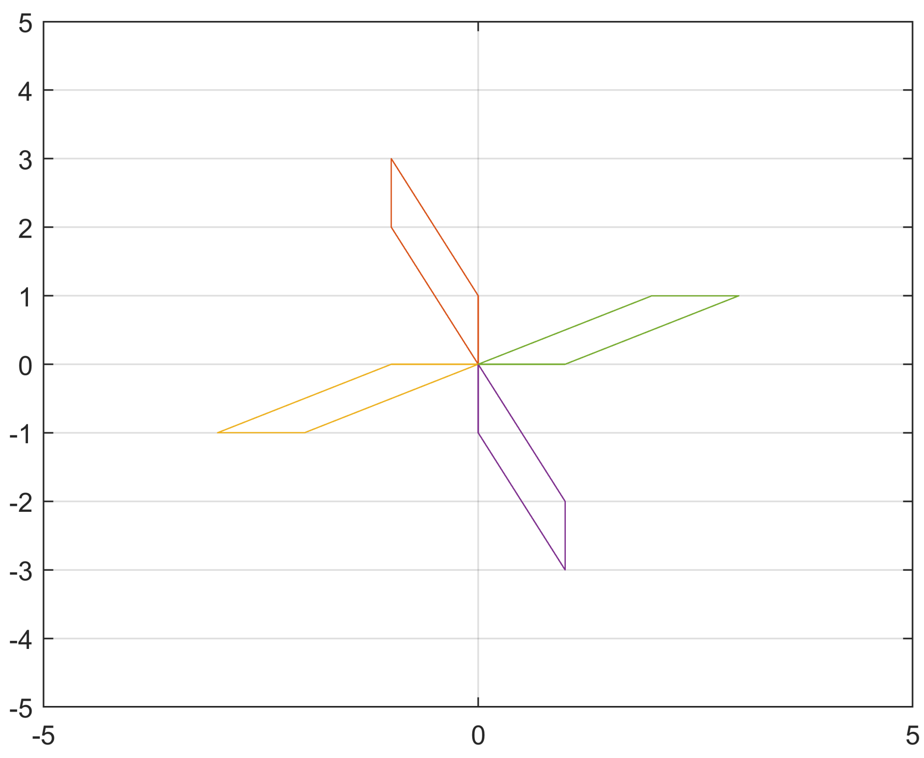
```
function [] = rotation(Q,n)
hold on
grid on
theta=(2*pi)/n;
i = 1;
j = 2;
m = 2;
A=givens(m,i,j,theta);
for k=1:n
    Q=transf(A,Q);
end
hold off
```

```
HS=[1 2;0 1]
```

```
HS = 2×2
     1     2
     0     1
```

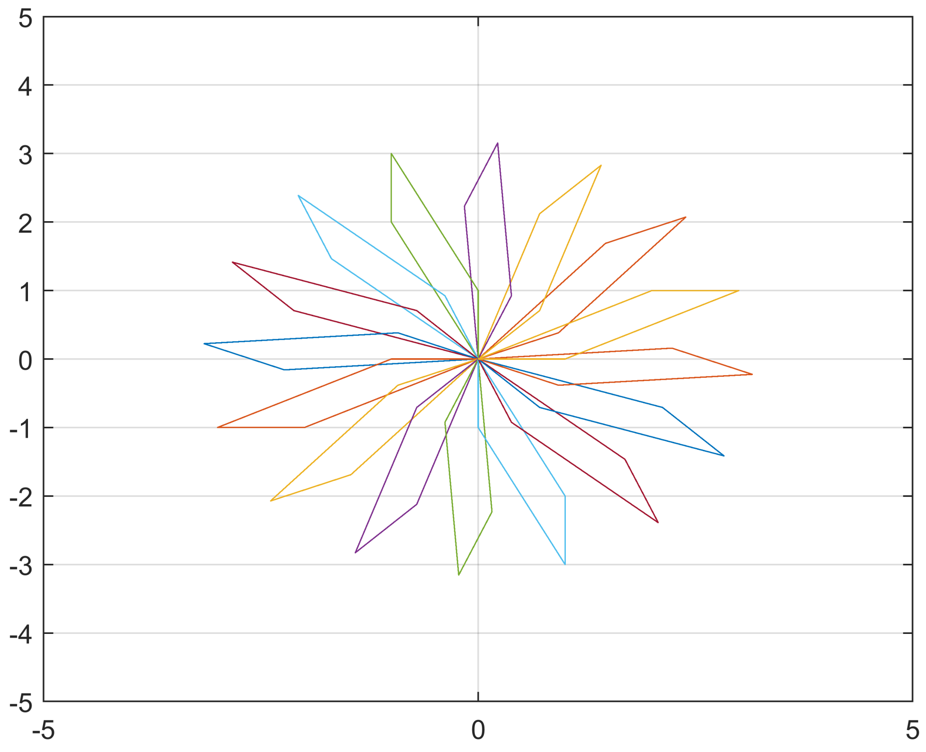
```
A=HS;
P=[0 1 1 0 0;0 0 1 1 0];
%(a)
Q=transf(A,P);
n=4;
rotation(Q,n)
```

```
G = 2×2
     1     0
     0     1
the Givens rotation matrix G is
    0.0000   -1.0000
    1.0000    0.0000
```



```
%(b)
Q=transf(A,P);
n=16;
rotation(Q,n)
```

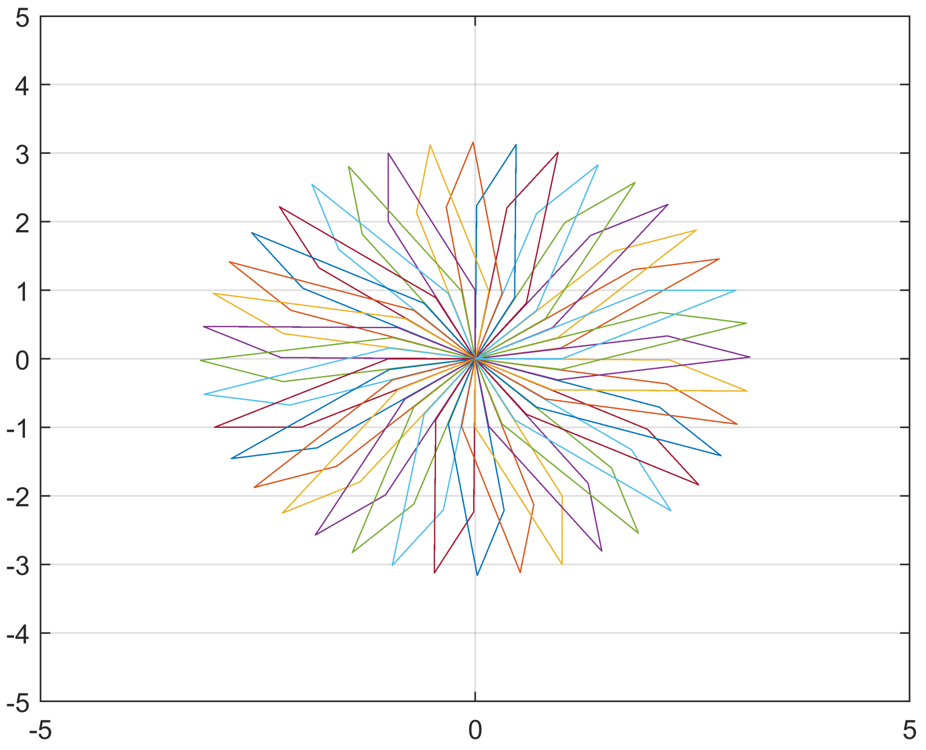
```
G = 2×2
     1     0
     0     1
the Givens rotation matrix G is
    0.9239   -0.3827
    0.3827    0.9239
```



```
%(c)
Q=transf(A,P);
n=40;
rotation(Q,n)
```

```
G = 2×2
     1     0
     0     1
the Givens rotation matrix G is
    0.9877   -0.1564
    0.1564    0.9877
```

```
hold off
```


5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
0
5

Exercise 6

```
type inverse.m
```

```
function B = inverse(a,C,D)
B = [];
if(D == 0)
    disp('Matrix A is not invertible');
    return;
end
B = (1/D)*(C')
F = inv(a);
if(closetozeroroundoff(B-F,7)~=0)
    disp('Whats wrong with B and F');
else
    disp('The inverse is calculated correctly and it is a matrix');
end
end
```

```
type determine.m
```

```
function D=determine(a,C)
D=[];
n=size(a,1);
if(rank(a)~=n)
    disp('the determinant of the matrix a is')
    D=0
    return;
end
E=zeros(n,2);
for i=1:n
    for j=1:n
        E(i) = E(i) + (a(i,j) * C(i,j));
    end
end
d=det(a);
if(closetozeroroundoff(d-E(1,1),7)==0)
    D=E(1,1);
    disp('the determinant of the matrix a is')
    D
else
    disp('Something went wrong!')
    D=[];
end
end
```

```
type cofactor.m
```

```
function C=cofactor(a)
[m,n]=size(a);
C=zeros(m,n);
for i=1:m;
    for j=1:n;
        aij=a;
        aij(i,:)=[];
        aij(:,j)=[];
        C(i,j)=(-1)^(i+j)*det(aij);
    end
end
disp('the matrix of cofactors of a is')
C
end
```

```
type closetozeroroundoff.m
```

```
function B=closetozeroroundoff(A,p)
A(abs(A)<10^-p)=0;
B=A;
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
C = 5×5
   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 a is
D = 120
```

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

```
B = 5×5
    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 a matrix
```

```
%(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
C = 4×4
     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
D = 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
C = 3×3
   -53    22     7
    52    -8   -68
   -23   -38    37
```

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

```
the determinant of the matrix a is
D = -360
```

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

```
B = 3×3
    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 a matrix
```

```
%(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
C = 4×4
10^3 ×
   -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
D = 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
C = 4×4
    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 a is
D = 1.6534e-07
```

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

```
B = 4×4
10^3 ×
    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 a matrix
```

```
%(f)
a=rand(2)
```

```
a = 2×2
    0.5383    0.0782
    0.9961    0.4427
```

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

```
the matrix of cofactors of a is
C = 2×2
    0.4427   -0.9961
   -0.0782    0.5383
```

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

```
the determinant of the matrix a is
D = 0.1604
```

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

```
B = 2×2
    2.7592   -0.4873
   -6.2088    3.3554
The inverse is calculated correctly and it is a matrix
```

BONUS

```
S=a;
original=S(1,1);
S(1,1)=S(2,2);
S(2,2)=original;
S(1,2)=-S(1,2);
S(2,1)=-S(2,1);
if(S==C')
    disp('')
    disp('the method of calculating the inverse of a 2x2 matrix is agreeable with the method o
    disp('the inverse matrix using cofactors.')
end
```

```
the method of calculating the inverse of a 2x2 matrix is agreeable with the method of calculating
the inverse matrix using cofactors.
```

Exercise 7

```
type production.m
```

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

nonnegative = 0;
greaterthanone = 0;

for i = 1:n
    for j = 1:n
        if C(i, j) < 0
          nonnegative = 1;
            break
        end
    end
end


for i = 1:n
    sum = 0;
    col = C(:, i);
    for j = 1:n
        sum = sum + col(j, 1);
    end
    if sum >1
        greaterthanone = 1;
        break
    end
end

if nonnegative
    disp('C contains negative entries.')
end
if greaterthanone
    disp('The sum of one or more columns in C are greater than or equal to 1.')
end
if(nonnegative || greaterthanone)
    return
end

    x =  inv(eye(n)-C) * d;
    nonfeasible = 0;
    for i = 1:n
        col = x(i,:);
        for j = 1:size(col, 1)
            if col(j, 1) < 1
                nonfeasible = 1;
                break
            end
        end
    end

    if nonfeasible
        disp('check the code!')
        x = []
        return
    end

    if ~nonfeasible
```

```
        disp('the unique product vector is ')
        x
    end
    x0 = d;
    k = 1;

    while any(closetozeroroundoff(x-x0, 1))
       x0 = d + C*x0;
       k = k+1;
    end
    x1 = x0;


    disp('the product vector calculated by recurrence relation is')
    x1

    fprintf('the number of iterations to match the output is %i\n' , k)


end
```

```
type closetozeroroundoff.m
```

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

```
%(a)
C = [.5 .4 .2 ; .2 .3 .1 ; .1 .1 .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 product vector is
x = 3×1
  225.9259
  118.5185
   77.7778
the product vector calculated by recurrence relation is
x1 = 3×1
  225.8298
  118.4705
   77.7467
the number of iterations to match the output is 29
```

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

```
C = 7×7
    0.1588    0.0064    0.0025    0.0304    0.0014    0.0083    0.1594
    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 product vector is
x = 7×1
10^5 ×
    0.9958
    0.9770
    0.5123
    1.3157
    0.4949
    3.2955
    0.1384
the product vector calculated by recurrence relation is
x1 = 7×1
10^5 ×
    0.9958
    0.9770
    0.5123
    1.3157
    0.4949
    3.2955
    0.1384
the number of iterations to match the output is 23
```

The (4, 3) entry of matrix C represents the output of the basic metal products and mining sector that is purchased by the basic nonmetal products and agriculture sector.

Sector 6 must produce 329550 units to satisfy the demand for its production.

```
%(c)
C = [.5 .4 .2; .2 .3 .1; -.1 .1 .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);
```

```
C contains negative entries.
```

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

```
C = 7×7
    0.1588    0.0064    0.0025    0.0304    0.0014    0.0083    0.1594
    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 product vector is
x = 7×1
10^5 ×
    1.3403
    1.3169
    0.6947
    1.7691
    0.6660
    4.4377
    0.1843
the product vector calculated by recurrence relation is
x1 = 7×1
10^5 ×
    1.3403
    1.3169
    0.6947
    1.7691
    0.6660
    4.4377
    0.1843
the number of iterations to match the output is 24
```

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

```
C = 7×7
    1.1588    0.0064    0.0025    0.0304    0.0014    0.0083    0.1594
    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 sum of one or more columns in C are greater than or equal to 1.
```

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

```
C = 7×7
    1.1588    0.0064    0.0025    0.0304    0.0014    0.0083    0.1594
    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);
```

```
The sum of one or more columns in C are greater than or equal to 1.
```