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 # 12

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

1. James Dika

2. Erin Potter

3. David Rosenbaum

4. Christian Fowler

5. Spencer Dupee

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

```
%Exercise1
type ele1

function [ E1 ] = ele1( n, r, i, j )

E1 = eye(n);
E1(j,:)= E1(j,:)+E1(i,:)*r;

end

type ele2

function [ E2 ] = ele2( n, i, j )

E2 = eye(n);
E2([i, j], :) = E2([j, i], :);

end

type ele3

function [ E3 ] = ele3( n, j, k )

E3=eye(n);
E3(j, :)=E3(j, :)*k;

end

type closetozeroroundoff

function [ B ] = closetozeroroundoff( A )

[m,n]=size(A);
for i=1:m
      for j=1:n
            if abs(A(i,j))<10^(-7)
                  A(i,j) = 0;
        end %if
      end
end
B=A;
end %function

format compact
format rat
A=[2 1 3; 1 0 2; 2 3 -1]
A =
       2      1      3
       1      0      2
       2      3      -1
E2=ele2(3, 1, 2)
E2 =
       0      1      0
       1      0      0
       0      0      1
A1=E2*A
A1 =
       1      0      2
```

```
       2     1     3
       2     3     -1
E1=ele1(3, -2, 1, 2)
E1 =
       1     0     0
      -2     1     0
       0     0     1
A2=E1*A1
A2 =
       1     0     2
       0     1     -1
       2     3     -1
E1=ele1(3, -2, 1, 3)
E1 =
       1     0     0
       0     1     0
      -2     0     1
A3=E1*A2
A3 =
       1     0     2
       0     1    -1
       0     3    -5
E1=ele1(3, -3, 2, 3)
E1 =
       1     0     0
       0     1     0
       0     -3    1
A4=E1*A3
A4 =
       1     0     2
       0     1     -1
       0     0     -2
E1=ele1(3,1,3,1)
E1 =
       1     0     1
       0     1     0
       0     0     1
A5=E1*A4
A5 =
       1     0     0
       0     1    -1
       0     0    -2
E3=ele3(3, 3, -1/2)
E3 =
       1     0     0
       0     1     0
       0     0     -1/2
A6=E3*A5
A6 =
       1     0     0
       0     1    -1
       0     0     1
E1=ele1(3, 1, 3, 2)
E1 =
       1     0     0
       0     1     1
       0     0     1
A7=E1*A6
```

```
A7 =
     1     0     0
     0     1     0
     0     0     1
```

```
%Excerise2
type inverses

function [ D ] = inverses( A )

[m, n] = size( A )
if m~= n
    D = [];
    disp('Matrix A is not invertible')
    return
end %if
if rank( A ) < m
    D = [];
    disp('Matrix A is not invertible')
    return
end %if
B = [A eye(n)];
C = rref( B );
for i = 1:n
    C(: , 1) = []
end %for
D = C;
end %function


%a
A = [4 0 -7 -7; -6 1 11 9; 7 -5 10 19; -1 2 3 -1]
A =
     4     0    -7    -7
    -6     1    11     9
     7    -5    10    19
    -1     2     3    -1
D = inverses(A)
m =
     4
n =
     4
C =
     0     0     0   -19   -14     0     7
     1     0     0  -549  -401    -2   196
     0     1     0   267   195     1   -95
     0     0     1  -278  -203    -1    99
C =
     0     0   -19   -14     0     7
     0     0  -549  -401    -2   196
     1     0   267   195     1   -95
     0     1  -278  -203    -1    99
C =
     0   -19   -14     0     7
     0  -549  -401    -2   196
     0   267   195     1   -95
     1  -278  -203    -1    99
C =
   -19   -14     0     7
  -549  -401    -2   196
   267   195     1   -95
  -278  -203    -1    99
D =
```

```
   -19   -14     0     7
  -549  -401    -2   196
   267   195     1   -95
  -278  -203    -1    99

%b
A = [1 -3 2 -4; -3 9 -1 5; 2 -6 4 -3; -4 12 2 7]
A =
     1    -3     2    -4
    -3     9    -1     5
     2    -6     4    -3
    -4    12     2     7
D = inverses(A)
m =
     4
n =
     4
Matrix A is not invertible
D =
     []

%c
A = magic(5)
A =
    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
D = inverses(A)
m =
     5
n =
     5
C =
         0         0         0         0   -0.0049    0.0512   -0.0354
0.0012    0.0034
    1.0000         0         0         0    0.0431   -0.0373   -0.0046
0.0127    0.0015
         0    1.0000         0         0   -0.0303    0.0031    0.0031
0.0031    0.0364
         0         0    1.0000         0    0.0047   -0.0065    0.0108
0.0435   -0.0370
         0         0         0    1.0000    0.0028    0.0050    0.0415
-0.0450    0.0111
C =
         0         0         0   -0.0049    0.0512   -0.0354    0.0012
0.0034
         0         0         0    0.0431   -0.0373   -0.0046    0.0127
0.0015
    1.0000         0         0   -0.0303    0.0031    0.0031    0.0031
0.0364
         0    1.0000         0    0.0047   -0.0065    0.0108    0.0435
-0.0370
         0         0    1.0000    0.0028    0.0050    0.0415   -0.0450
0.0111
C =
         0         0   -0.0049    0.0512   -0.0354    0.0012    0.0034
```

```
         0         0    0.0431   -0.0373   -0.0046    0.0127    0.0015
         0         0   -0.0303    0.0031    0.0031    0.0031    0.0364
    1.0000         0    0.0047   -0.0065    0.0108    0.0435   -0.0370
         0    1.0000    0.0028    0.0050    0.0415   -0.0450    0.0111
C =
         0   -0.0049    0.0512   -0.0354    0.0012    0.0034
         0    0.0431   -0.0373   -0.0046    0.0127    0.0015
         0   -0.0303    0.0031    0.0031    0.0031    0.0364
         0    0.0047   -0.0065    0.0108    0.0435   -0.0370
    1.0000    0.0028    0.0050    0.0415   -0.0450    0.0111
C =
   -0.0049    0.0512   -0.0354    0.0012    0.0034
    0.0431   -0.0373   -0.0046    0.0127    0.0015
   -0.0303    0.0031    0.0031    0.0031    0.0364
    0.0047   -0.0065    0.0108    0.0435   -0.0370
    0.0028    0.0050    0.0415   -0.0450    0.0111
D =
   -0.0049    0.0512   -0.0354    0.0012    0.0034
    0.0431   -0.0373   -0.0046    0.0127    0.0015
   -0.0303    0.0031    0.0031    0.0031    0.0364
    0.0047   -0.0065    0.0108    0.0435   -0.0370
    0.0028    0.0050    0.0415   -0.0450    0.0111
D = inv(A)
D =
   -0.0049    0.0512   -0.0354    0.0012    0.0034
    0.0431   -0.0373   -0.0046    0.0127    0.0015
   -0.0303    0.0031    0.0031    0.0031    0.0364
    0.0047   -0.0065    0.0108    0.0435   -0.0370
    0.0028    0.0050    0.0415   -0.0450    0.0111

%d
A = magic(4)
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
D = inverses(A)
m =
     4
n =
     4
Matrix A is not invertible
D =
     []
D = inv(A)
[_Warning: Matrix is close to singular or badly scaled. Results may be
inaccurate. RCOND =  1.306145e-17.]_
D =
   1.0e+14 *
    0.9382    2.8147   -2.8147   -0.9382
    2.8147    8.4442   -8.4442   -2.8147
   -2.8147   -8.4442    8.4442    2.8147
   -0.9382   -2.8147    2.8147    0.9382
%Matlab runs the numbers too early. The results obtained by using inv
function is inaccurate for part d because the function rounds the numbers to
find the determinant. Then, supposing that an inverse matrix could be made,
Matlab finds a similar inverse to the real inverse matrix.
```

```
%Exercise 3
type solvesys

function [ C,N ] = solvesys( A )
%This function will solve a matrix equation
%given an input nXn matrix, A , and a predetermined
%b matrix using 3 different methods.
%Output is an nX3 matrix,C, of the solutions
%calculated in the three different ways
%and a column vector, N, of the error between
%each method.

[n,n] = size(A);
b = fix(10*rand(n,1))
format long
if det(A) == 0
    disp('The system is either inconsistent')
    disp('or the solution is not unique.')
    C = [];
    N = [];
else
  %Backslash method
  C_back = A\b;



  %Inverse method
  C_inv = inv(A)*b;
```

```
  %Rref method
  sol = rref([A b]);
  C_rref = [];
  for ii = 1: size(sol)
  C_rref(ii,:) = sol(ii,end);
  end

end
C = [C_back C_inv C_rref]
N = [norm(C(:,1)-C(:,2));norm(C(:,2)-C(:,3));norm(C(:,3)-C(:,
1))]

end

%part a
A = magic(6)
A =
    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11
solvesys(A);
b =
     7
     3
```

```
     2
     4
     0
     1

[Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND =  4.800964e-18.]

C =
   1.0e+14 *
  Columns 1 through 2
  -7.505999378950825  -7.505999378950825
  -7.505999378950825  -7.505999378950830
   3.752999689475411   3.752999689475409
   7.505999378950829   7.505999378950830
   7.505999378950821   7.505999378950818
  -3.752999689475411  -3.752999689475411
  Column 3
                   0
                   0
                   0
                   0
                   0
   0.000000000000010
N =
   1.0e+15 *
   0.000000000000001
   1.592262918131443
```

```
   1.592262918131443

%Part b
A = eye(5)
A =
     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
solvesys(A);
b =
     9
     9
     5
     0
     2
C =
     9     9     9
     9     9     9
     5     5     5
     0     0     0
     2     2     2
N =
     0
     0
     0
%Each column in the matrix C is equal to the vector
```

```
%b since A is a 5X5 identity matrix.

%Part c
A = randi(20,4,4)
A =
     8     4    10     4
    17    13    11    14
     1    15     6     4
     1    13    15     8
solvesys(A);
b =
     6
     7
     0
     9
C =
  Columns 1 through 2
  -0.157971014492754  -0.157971014492754
  -0.401449275362319  -0.401449275362319
   0.672463768115942   0.672463768115942
   0.536231884057971   0.536231884057971
  Column 3
  -0.157971014492754
  -0.401449275362319
   0.672463768115942
   0.536231884057971
N =
   1.0e-15 *
```

```
   0.337661150723213
   0.124126707662364
   0.271947991102104


%Part d
A = magic(3)
A =
     8     1     6
     3     5     7
     4     9     2
solvesys(A);
b =
     7
     4
     4
C =
  Columns 1 through 2
   0.708333333333333   0.708333333333333
   0.083333333333333   0.083333333333333
   0.208333333333333   0.208333333333333
  Column 3
   0.708333333333333
   0.083333333333333
   0.208333333333333
N =
   1.0e-15 *
   0.117756934401283
```

```
   0.115277563368905

   0.013877787807814


%Part e

format rat,A = hilb(6)

A =

  Columns 1 through 3

       1              1/2            1/3

       1/2            1/3            1/4

       1/3            1/4            1/5

       1/4            1/5            1/6

       1/5            1/6            1/7

       1/6            1/7            1/8

  Columns 4 through 6

       1/4            1/5            1/6

       1/5            1/6            1/7

       1/6            1/7            1/8

       1/7            1/8            1/9

       1/8            1/9            1/10

       1/9            1/10           1/11

solvesys(A);

b =

       4

       3

       5

       5

       8

       7
```

```
C =

   1.0e+07 *

  Columns 1 through 2

   0.001833000000232   0.001833000000129
  -0.052290000006917  -0.052290000003871
   0.353640000048100   0.353640000026992
  -0.919296000127467  -0.919296000071650
   1.013922000142612   1.013922000080257
  -0.399168000056772  -0.399168000031978

  Column 3

   0.001833000000000
  -0.052290000000000
   0.353640000000000
  -0.919296000000000
   1.013922000000000
  -0.399168000000000

N =

   0.000898514234412
   0.001155035187572
   0.002053548401709

%The values in the matrix N are so much larger for the
%hilbert matrix because the solutions in C are
%of such a large order of magnitude greater than
%the solutions in previous parts.
```

```
%Exercise 4

type arevol


function D = arevol( B )

%Calculates the area of a parallelogram or parallelepiped

if (size(B, 2) == 3)

    v1 = B(:,2) - B(:,1);

    v2 = B(:,3) - B(:,1);

    A = [v1 v2];

else

    v1 = B(:,2) - B(:,1);

    v2 = B(:,3) - B(:,1);

    v3 = B(:,4) - B(:,1);

    A = [v1 v2 v3];

end


D = abs(det(A));

D = closetozeroroundoff(D);


if (D == 0)

    if (size(A, 2) == 2)

        disp('The points lie on the same line and no
parallelogram can be built');

    else

        disp('The points lie in the same plane and no
parallelepiped can be built');

    end

else

    if (size(A, 2) == 2)
```

```
        fprintf('The area of the parallelogram is %d', D);
    else
        fprintf('The volume of the parallelepiped is %d', D);
    end
end
end

B = randi([-10,10], 2, 3)


B =
     7    -8     3
     9     9    -8

D=arevol(B)
The area of the parallelogram is 255
D =
   255

B = randi([-10,10], 3, 4)


B =
    -5    10    10    -8
     1    -7     0    -2
    10    10     6     9

D=arevol(B)
The volume of the parallelepiped is 3.810000e+02
D =
```

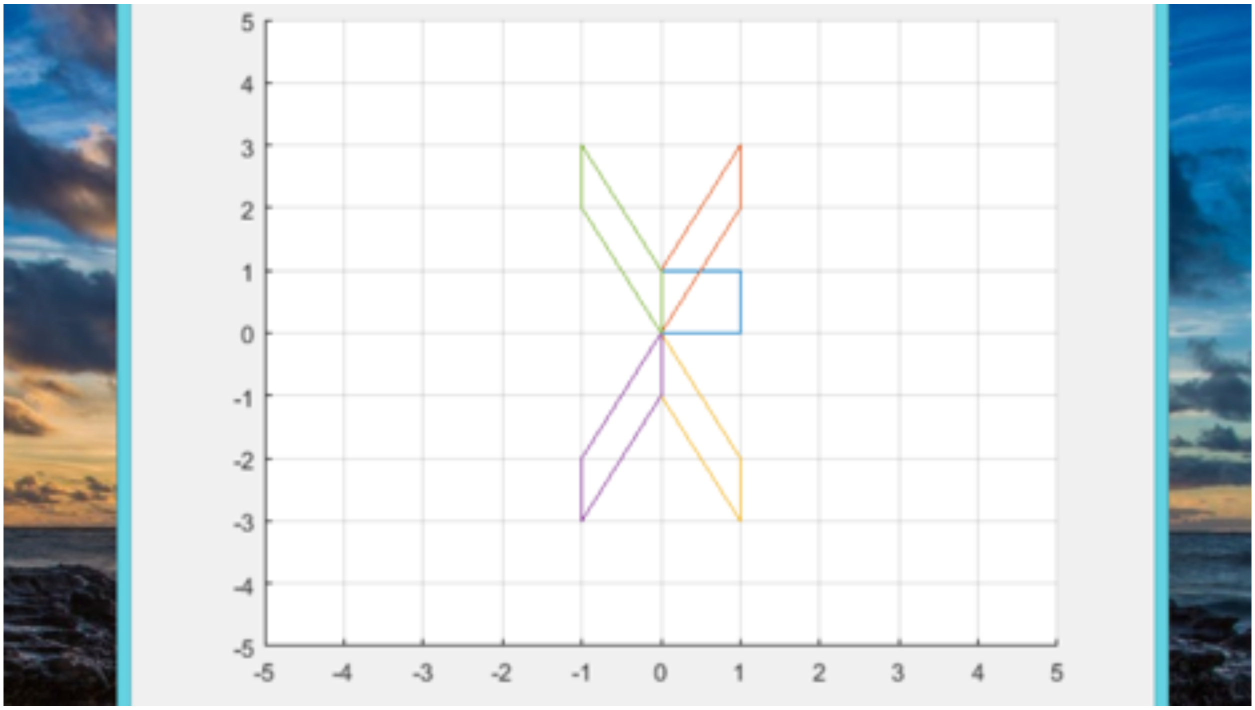
```
  381.0000


X = randi([-10,10], 2, 1), B = [X, -X, 2*X]


X =

     6

    10




B =

     6    -6    12

    10   -10    20


D=arevol(B)

The points lie on the same line and no parallelogram can be
built


D =

     0


>> D=arevol(X)

Index exceeds matrix dimensions.


Error in arevol (line 8)

    v1 = B(:,2) - B(:,1);


X = randi([-10,10], 3, 1), Y = randi([-10,10], 3, 1), B = [X,
Y , X + Y , X - Y]
```

```
X =

     3
   -10
     7


Y =

     9
     4
     5


B =

     3     9    12    -6
   -10     4    -6   -14
     7     5    12     2

D=arevol(X)
Index exceeds matrix dimensions.

Error in arevol (line 8)
    v1 = B(:,2) - B(:,1);

D=arevol(Y)
Index exceeds matrix dimensions.

Error in arevol (line 8)
    v1 = B(:,2) - B(:,1);
```

```
D=arevol(B)

The points lie in the same plane and no parallelepiped can be
built


D =

     0
```

```
%Exercise 5
R1 = [1 0;0 -1];
R2 = [-1 0;0 1];
VS = [1 0;2 1];
type transf
function C = transf(A,E)
E=A*E; % Multiplies matrices A and E
x=E(1,:); % x is the first row of E
y=E(2,:); % y is the second row of E
plot(x,y) % plots x and y values on xy plane
v=[-5 5 -5 5]; % creates the vector v
axis(v) % sets x and y axis values to start at -5 and end at 5
(values in vector v)
grid % either displays or turns off gridlines depending on
current setting
C=E; % C is set equal to E
grid % Removes gridlines or displays gridlines depending on
current setting
end

E=[0 1 1 0 0; 0 0 1 1 0]
E =
0 1 1 0 0
0 0 1 1 0

A=eye(2)
A =
1 0
0 1

hold
Current plot held

grid

C = transf(A,E)
C =
0 1 1 0 0
0 0 1 1 0

E = C;

A = VS
A =
1 0
2 1
```

```
C = transf(A,E)
C =
0 1 1 0 0
0 2 3 1 0

E = C;

A = R1
A =
1 0
0 -1

C = transf(A,E)
C =
0 1 1 0 0
0 -2 -3 -1 0

E = C;

A = R2
A =
-1 0
0 1

C = transf(A, E)
C =
0 -1 -1 0 0
0 -2 -3 -1 0

E = C;

A = R1
A =
1 0
0 -1

A = transf(A, E)
A =
0 -1 -1 0 0
0 2 3 1 0
```


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

```
%Exercise 6

function C = cofactor(a)
%Outputs the cofactor matrix of matrix a
%   creates minor matricies of size (n-1)x(n-1) with current row i and
column j removed.
%   sets the output, C, to a matrix in which entry i,j =
%   [(-1)^(i+j)]*det(minor).

[m,n] = size(a);
for i=1:m
    for j=1:n
        y = a;              %copies a,
        y([i],:) = [];      %deletes row i,
        y(:,[j]) = [];      %deletes row j,
        C(i,j) = det(y)*(-1)^(i+j);     %inputs cofactor entry.
        clear('y');
    end
end
end

type determine

function D = determine(a,C)
%The function calculates the cofactor expansion across each row of a -
they form entries of a
%vector D1; and it calculates the cofactor expansion down each column of a
- they form
%entries of the vector D2.
%   constructs D1 and D2 according to the size of a, then fills in with
%   corresponding cofactor expansion: sum of a(i,j)*C(i,j) across row/
column.
format rat;
[m,n] = size(a);
D1 = zeros(m,1);
D2 = zeros(1,n);
for i=1:m
    for j=1:n
        D1(i,1) = D1(i,1) + a(i,j)*C(i,j);
        D2(1,i) = D2(1,i) + a(j,i)*C(j,i);
    end
end
for i = D1
    if abs(i-D2) > 1e-7         %confirms all entries are the same
        disp('theres an error with my code');
        D = [];
        return;
    end
end
if D1(1) < 1e-7
    D1(1) = 0;                  %corrects for rounding errors.
end
```

```
D = D1(1);
end

type inverse

function B = inverse(a,C,D)
%Inverts matrix a. Inputs C & D are the cofactor matrix of a and the
determinant of a respectively.
%   Sets each element of B = (1/D)*transpose(C), assuming B is not
singular.
[m,n] = size(a);
if rank(a) < m  %rank because det(a) could be very close but not exactly
zero due to rounding.
    B = [];
else
    B = (1/D)*transpose(C);

end

format rat
a = diag([1,2,3,4])
a =
       1              0              0              0
       0              2              0              0
       0              0              3              0
       0              0              0              4
C = cofactor(a)
C =
      24              0              0              0
       0             12              0              0
       0              0              8              0
       0              0              0              6
D = determine(a,C)
D =
      24
det(a)
ans =
      24
%they are the same.
B = inverse(a,C,D)
B =
       1              0              0              0
       0              1/2            0              0
       0              0              1/3            0
       0              0              0              1/4
inv(a)
ans =
       1              0              0              0
       0              1/2            0              0
       0              0              1/3            0
       0              0              0              1/4
%they are the same.
a = ones(5)
a =
```

```
  Columns 1 through 4
       1              1              1              1
       1              1              1              1
       1              1              1              1
       1              1              1              1
       1              1              1              1
  Column 5
       1
       1
       1
       1
       1
C = cofactor(a)
C =
  Columns 1 through 4
       0              0              0              0
       0              0              0              0
       0              0              0              0
       0              0              0              0
       0              0              0              0
  Column 5
       0
       0
       0
       0
       0
D = determine(a,C)
D =
       0
det(a)
ans =
       0
%they are the same.
B = inverse(a,C,D)
B =
     []
inv(a)
[_Warning: Matrix is singular to working precision.]_
ans =
  Columns 1 through 4
       1/0            1/0            1/0            1/0
       1/0            1/0            1/0            1/0
       1/0            1/0            1/0            1/0
       1/0            1/0            1/0            1/0
       1/0            1/0            1/0            1/0
  Column 5
       1/0
       1/0
       1/0
       1/0
       1/0
%My program is strictly superior.
a = magic(5)
a =
```

```
  Columns 1 through 4
      17             24              1              8
      23              5              7             14
       4              6             13             20
      10             12             19             21
      11             18             25              2
  Column 5
      15
      16
      22
       3
       9
C = cofactor(a)
C =
  Columns 1 through 4
  -25025         218725        -153400          23725
  259350        -189150          15600         -33150
 -179400         -23400          15600          54600
    5850          64350          15600         220350
   17225           7475         184600        -187525
  Column 5
   13975
   25350
  210600
 -228150
   56225
D = determine(a,C)
D =
 5070000
det(a)
ans =
 5070000
%they are the same.
B = inverse(a,C,D)
B =
  Columns 1 through 4
     -77/15600      133/2600       -23/650          3/2600
      89/2063       -97/2600        -3/650         33/2600
     -59/1950         1/325          1/325          1/325
      73/15600      -17/2600         7/650        113/2600
      43/15600        1/200         27/650         -9/200
  Column 5
      53/15600
      23/15600
      71/1950
    -577/15600
      98/8837
inv(a)
ans =
  Columns 1 through 4
     -77/15600      133/2600       -23/650          3/2600
      89/2063       -97/2600        -3/650         33/2600
     -59/1950         1/325          1/325          1/325
      73/15600      -17/2600         7/650        113/2600
```

```
      43/15600        1/200         27/650         -9/200
  Column 5
      53/15600
      23/15600
      71/1950
    -577/15600
      98/8837
%they are the same.
a = magic(4)
a =
      16              2              3             13
       5             11             10              8
       9              7              6             12
       4             14             15              1
C = cofactor(a)
C =
    -136           -408            408            136
    -408          -1224           1224            408
     408           1224          -1224           -408
     136            408           -408           -136
D = determine(a,C)
D =
       0
det(a)
ans =
      -1/689889648801
%My program used to give a similar answer but then I wrote an if statement
that corrects for determinants within 1e-7 of 0.
B = inverse(a,C,D)
B =
     []
inv(a)
[_Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND =  1.306145e-17.]_
ans =
       *              *              *              *
       *              *              *              *
       *              *              *              *
       *              *              *              *
%My program checks for singularity before computation.
a = hilb(4)
a =
       1              1/2            1/3            1/4
       1/2            1/3            1/4            1/5
       1/3            1/4            1/5            1/6
       1/4            1/5            1/6            1/7
C = cofactor(a)
C =
       1/378000      -1/50400        1/25200       -1/43200
      -1/50400        1/5040        -1/2240         1/3600
       1/25200       -1/2240         3/2800        -1/1440
      -1/43200        1/3600        -1/1440         1/2160
D = determine(a,C)
D =
```

```
        1/6048000
det(a)
ans =
        1/6048000
%they are the same.
B = inverse(a,C,D)
B =
      16            -120              240             -140
    -120            1200            -2700             1680
     240           -2700             6480            -4200
    -140            1680            -4200             2800
inv(a)
ans =
      16            -120              240             -140
    -120            1200            -2700             1680
     240           -2700             6480            -4200
    -140            1680            -4200             2800
%they are the same.
```