```
%  CS412 FAll 2002
%  HW3
%  Data Fok
%  ---------------------------------------------------------------------

% Question 1

The Matlab scripts are as follow:
-----------------------------------------------------------------------

%  get the coordinates of the joints.
%  xyz a matrix with the data in Connections.dat.

xyz = load('Coordinates.dat') ;

%  get the information about which joints the members connect.
%  for example, member 1 connects joints 1 and 4.
%  connect is a matrix with the data in Connections.dat.

connect = load('Connections.dat') ;

members = size(connect,1) ;  % # of members

%  delcare a matix to hold the direction cosine
DC = zeros(33,27);    % matrix of the Direction Cosine

%  calculate the direction cosine of the joints in each member
%  put the direction cosine in the correction location of the matrix

for m = 1:members

    %  Get the two joints at the ends of the member.
    joint1 = connect(m,1) ;
    joint2 = connect(m,2) ;
    %  Get the coordinates of points (x1,y1,z1) and (x2,y2,z2)
    Xi = xyz(joint1, 1 ) ;
    Yi = xyz(joint1, 2 ) ;
    Zi = xyz(joint1, 3 ) ;
    Xj = xyz(joint2, 1 ) ;
    Yj = xyz(joint2, 2 ) ;
    Zj = xyz(joint2, 3 ) ;

    directioncosineX = (Xj - Xi) / sqrt((Xj - Xi)^2 + (Yj - Yi)^2 + (Zj - Zi)^2);
    directioncosineY = (Yj - Yi) / sqrt((Xj - Xi)^2 + (Yj - Yi)^2 + (Zj - Zi)^2);
    directioncosineZ = (Zj - Zi) / sqrt((Xj - Xi)^2 + (Yj - Yi)^2 + (Zj - Zi)^2);

    % insert the value to the right place in the matrix

    % filling row x for joint 1
    DC( (1 + (joint1 - 1)*3),  m) = directioncosineX;
    % filling row y for joint 1
    DC( (2 + (joint1 - 1)*3),  m) = directioncosineY;
    % filling row z for joint 1
    DC( (3 + (joint1 - 1)*3),  m) = directioncosineZ;

    % filling row x for joint 2
    DC( (1 + (joint2 - 1)*3),  m) = - directioncosineX;
    % filling row y for joint 2
    DC( (2 + (joint2 - 1)*3),  m) = - directioncosineY;
    % filling row z for joint 2
    DC( (3 + (joint2 - 1)*3),  m) = - directioncosineZ;

end % by this the Direction matrix with 33 rows and 27 cols is formed
```

```
% now elementing the row y and z for joint #1, and row x and Z for joint
% #2,3

% Formating the final 27 x 27 matrix
DCF = zeros(27,27);

DCF(1,:) = DC(1,:);
DCF(2,:) = DC(5,:);
DCF(3,:) = DC(8,:);
DCF(4:27,:) = DC(10:33,:);

% Forming the load matrix with different given Load at joint 11 (x, y, z)
LOAD = zeros(27,1);

LOAD(25:27,1) = [0.0;0.0;10] ;

% solving the equation
fm = DCF\LOAD;

%  ------------------------------------------------------------Script ends
```

Comment on the results:
a) All the force on members (fm) do not exceed the capabitily of the structure.

```
fm =

    6.7364
    1.0765
    1.0765
   -2.2521
   -4.4753
   -2.9251
   -2.9251
   -4.4753
   -2.2521
   -4.9354
   -4.9354
   -1.8382
   -7.5538
   11.9140
    5.2417
    5.2417
   11.9140
   -7.5538
   10.0671
   -1.6815
   -4.0047
   -9.0978
   -4.0047
   -1.6815
   22.1169
  -13.2886
  -13.2886
```

b) fm(13), fm(20), fm(23) exceed the capcability of the structure

```
fm =

    5.3891
   -3.3461
    5.0686
  -16.8413
   10.9177
    2.2094
```

```
   -6.8895
  -18.0782
   13.2379
   -3.9483
   -3.9483
   -1.4705
  -24.4683
   -4.6489
   23.5855
  -15.1988
   23.7113
   12.3822
    8.0537
  -24.0721
   23.4109
   -7.2782
  -29.8184
   21.3817
   17.6935
   -7.9739
  -13.2878
```

c) All fm do not exceed the capabitily of the structure.

```
  fm =

    3.7452
    0.5985
    0.5985
    0.4004
   -4.5683
   -2.3905
   -2.3905
   -4.5683
    0.4004
   -2.7439
   -2.7439
   -3.4674
   -5.3053
   10.7924
    4.2837
    4.2837
   10.7924
   -5.3053
    5.5970
    1.0959
   -4.8529
   -7.4350
   -4.8529
    1.0959
   16.7443
  -11.2996
  -11.2996
```

d) fm(25) exceed the capcability of the structure

```
  fm =

    8.0837
    0.5906
    1.9930
   -5.2092
   -2.9541
   -2.7519
```

```
 -4.2684
 -7.7867
 -0.1960
 -5.9224
 -5.9224
 -2.2058
-12.1355
 11.9334
  9.5221
  3.0580
 16.6601
 -5.9937
 12.0806
 -5.8056
 -0.3698
-10.9173
 -9.2414
  1.7700
 26.5402
-15.5034
-16.3891

--------------------------------------------------end of Question 1

Question 2

 1) The order Matlab chose the rows to do the forward elimination is
    1 4       3       7       9       5       10      6       8       2

 2)

 L1 =

Columns 1 through 7

  1.0000         0         0         0         0         0         0
       0    1.0000         0         0         0         0         0
       0         0    1.0000         0         0         0         0
       0         0         0    1.0000         0         0         0
       0         0         0         0    1.0000         0         0
       0         0         0         0         0    1.0000         0
       0         0         0         0         0         0    1.0000
       0         0         0         0         0         0   -0.1805
       0         0         0         0         0         0   -0.4235
       0         0         0         0         0         0   -0.1313

Columns 8 through 10

       0         0         0
       0         0         0
       0         0         0
       0         0         0
       0         0         0
       0         0         0
       0         0         0
  1.0000         0         0
 -0.8769    1.0000         0
  0.2698    0.4352    1.0000

L2 =

Columns 1 through 7

  1.0000         0         0         0         0         0         0
```

```
         0    1.0000         0         0         0         0         0
         0         0    1.0000         0         0         0         0
         0         0         0    1.0000         0         0         0
         0         0         0    0.9527    1.0000         0         0
         0         0         0   -0.1573   -0.7911    1.0000         0
         0         0         0    0.1235   -1.0265    0.7585    1.0000
         0         0         0   -0.0123    0.3692   -0.9403         0
         0         0         0   -0.0073   -1.0481    1.0317         0
         0         0         0   -0.0015   -0.8476   -0.7024         0

  Columns 8 through 10

         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
    1.0000         0         0
         0    1.0000         0
         0         0    1.0000

  L3 =

  Columns 1 through 7

    1.0000         0         0         0         0         0         0
   -0.4805    1.0000         0         0         0         0         0
   -0.3715   -0.7503    1.0000         0         0         0         0
   -0.2155    0.1847   -0.9585    1.0000         0         0         0
   -0.1289   -0.3865   -1.4404         0    1.0000         0         0
    0.0146   -0.3999    0.3336         0         0    1.0000         0
   -0.0060   -0.2299   -0.6197         0         0         0    1.0000
    0.0004   -0.3196    0.0831         0         0         0         0
    0.0002    0.2377    0.0862         0         0         0         0
    0.0000   -0.0435    0.0213         0         0         0         0

  Columns 8 through 10

         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
    1.0000         0         0
         0    1.0000         0
         0         0    1.0000

P1 =

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

```
P2 =

     1     0     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     0     1

P3 =

     1     0     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     0     1

    so that
    L3 * P3 * L2 * P2 * L1 * P1 * B = U
    where U is in the upper trangular form.

3) RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.

   RANK(A,tol) is the number of diagonal elements of the matrix U
   in the LU factorization of A that are larger than tol.

   For U,

   tolerance =

    2.347858697756676e-015

  by viewing in long format

  U =

  Columns 1 through 3

   0.50000000000000   0.41421356237310   0.36602540378444
                  0   0.01947584945594   0.02844066543166
                  0                  0  -0.00061183691017
                  0                  0                  0
                  0                  0                  0
                  0                  0                  0
                  0                  0                  0
                  0                  0                  0
                  0                  0                  0

  Columns 4 through 6

   0.33333333333333   0.30901699437495   0.28989794855664
   0.03354422779337   0.03675737130053   0.03889923314793
  -0.00123450063688  -0.00178499263452  -0.00225809757179
   0.00000852042944   0.00002167523399   0.00003687502073
```

```
                0  -0.00000031548630  -0.00000097743382
                0                   0  -0.00000000252880
                0                   0                   0
                0                   0                   0
                0                   0                   0
                0                   0                   0

Columns 7 through 9

 0.27429188517743   0.26120387496374   0.25000000000000
 0.04037651288246   0.04141520024809   0.04215112349236
-0.00266319007867  -0.00301124546933  -0.00331196859308
 0.00005270989477   0.00006843494926   0.00008366486550
-0.00000192989480  -0.00000310320624  -0.00000443535313
-0.00000000924090  -0.00000002079085  -0.00000003719487
-0.00000000004647  -0.00000000019425  -0.00000000048753
                0  -0.00000000000024  -0.00000000000112
                0                   0   0.00000000000000
                0                   0                   0

Column 10

 0.24025307335204
 0.04267080449822
-0.00357336045232
 0.00009821040847
-0.00000587636375
-0.00000005812883
-0.00000000095550
-0.00000000000311
 0.00000000000002
 0.00000000000000

 we can see that there are only 8 diagonal elements of the matrix U
 in the LU factorization of A that are larger than tolerance. So when
 calling the rank(U) it return 8 even though it is a 10 x 10 matrix.
```