<table>
<thead>
<tr>
<th>Function</th>
<th>Variable Passed In</th>
<th>Size</th>
<th>Variable Passed Out</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ud_webvect.m</td>
<td>nele</td>
<td>1 by 1</td>
<td>WEBDIR</td>
<td>nele by 3</td>
</tr>
<tr>
<td></td>
<td>ends</td>
<td>nele by 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>beta_ang</td>
<td>nele by 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>coord</td>
<td>nnodes by 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ud_trans.m</td>
<td>del</td>
<td>1 by 3</td>
<td>T</td>
<td>12 by 12</td>
</tr>
<tr>
<td></td>
<td>webdir</td>
<td>1 by 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ud_order.m</td>
<td>nnodes</td>
<td>1 by 1</td>
<td>NFDOF</td>
<td>1 by 1</td>
</tr>
<tr>
<td></td>
<td>fixity</td>
<td>nnodes by 6</td>
<td>NSDOF</td>
<td>1 by 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NORDER</td>
<td>nnodes by 6</td>
</tr>
<tr>
<td>ud_pfdisps.m</td>
<td>nnodes</td>
<td>1 by 1</td>
<td>PF</td>
<td>NFDOF by 1</td>
</tr>
<tr>
<td></td>
<td>concen</td>
<td>nnodes by 6</td>
<td>DISP_S</td>
<td>NSDOF by 1</td>
</tr>
<tr>
<td></td>
<td>norder</td>
<td>nnodes by 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>fixity</td>
<td>nnodes by 6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
function `WEBDIR = ud_webvect(nele,ends,beta_ang,coord)`

Function purpose:
This function generates the element's current web unit vector.

Functions Called
<none>

Dictionary of Variables

Input Information:
`nele` == number of elements
`ends(i, 1:2)` == element `i`'s start and finish nodes
`ends(i, 1)` == start node for element `i`
`ends(i, 2)` == finish node for element `i`
`beta_ang(i)` == element `i`'s web rotation angle.

Note: MASTAN2 uses the following convention for defining a member's default web orientation:
A vector defining the element's local y-y axis with respect to the local coordinate system will have a non-zero component in the global Y-direction. If the element's local x-x axis (its length axis) is aligned with the global Y axis, then the element's local y-y axis is aligned with the global negative X axis.
After this initial orientation, element `i` may be rotated about its local x-x axis by the amount defined by its web rotation angle, `beta_ang(i)`. The angle is in RADIANS and assumes a right-hand convention with the local x-x axis running from the start node to the finish node.

Local Variables:
`x(1)` == i component in global space of element's local x-axis
`x(2)` == j component in global space of element's local x-axis
`x(3)` == k component in global space of element's local x-axis
`y(1)` == i component in global space of element's local y-axis
`y(2)` == j component in global space of element's local y-axis
`y(3)` == k component in global space of element's local y-axis
`z(1)` == i component in global space of element's local z-axis
`z(2)` == j component in global space of element's local z-axis
`z(3)` == k component in global space of element's local z-axis
`lvec` == length of temporary vector
`i` == loop index variable
`L` == element's length

Output Information:
`T(12,12)` == element's transformation matrix (row,col)

Good Luck CEE 361 Student!

function `T = ud_trans(del,webdir)`

Good Luck CEE 361 Student!

function `[ND0F,NSDOF,NORDER] = ud_order(nnodes,fixity)`

Function purpose:
This function defines the internal numbering for all of the DOFs in the structure.

Functions Called
<none>

Dictionary of Variables

Input Information:
`nnodes` == number of nodes
`fixity(i,6)` == restraint for node `i`'s 6 DOFs
NOTATION: NaN = free (f set)
# = supported (s set)
`fixity(i,1)` = displacement in X-direction for node `i`
`fixity(i,2)` = displacement in Y-direction for node `i`
`fixity(i,3)` = displacement in Z-direction for node `i`
function [PF, DISP_S] = ud_pfdisp(nnodes, concen, norder, fixity)

Function purpose:
This function defines the free DOF part of the external load vector (PF), and the prescribed displacement vector (DISP_S) which can include initial settlements!

Functions Called
<none>

Dictionary of Variables

Input Information:
nnodes == number of nodes
concen(1, 6) == concentrated load for node i's 6 DOFs
  concen(1, 1) = force in X-direction
  concen(1, 2) = force in Y-direction
  concen(1, 3) = force in Z-direction
  concen(1, 4) = moment about X-direction
  concen(1, 5) = moment about Y-direction
  concen(1, 6) = moment about Z-direction
norder(1, 6) == free and supported DOF #’s corresponding to node i’s 6 DOFs (dof,node)
  NOTE: norder will be positive if the DOF of the node is “free” and norder will be negative if the DOF of the node is “supported.”
fixity(1, 6) == prescribed displacement for node i’s 6 DOPs
  NOTATION: NaN = free (f set)
  # = supported (s set)
  fixity(1, 1) = displacement in X-direction
  fixity(1, 2) = displacement in Y-direction
  fixity(1, 3) = displacement in Z-direction
  fixity(1, 4) = rotation about X-direction
  fixity(1, 5) = rotation about Y-direction
  fixity(1, 6) = rotation about Z-direction

Local Variables:

node == node # being investigated
dof == DOF # being investigated
ivar == dummy variable

Output Information:
PF(i) == free DOF external load vector
DISP_S(i) == vector of supported DOF displacements

Good Luck CEE 361 Student!!