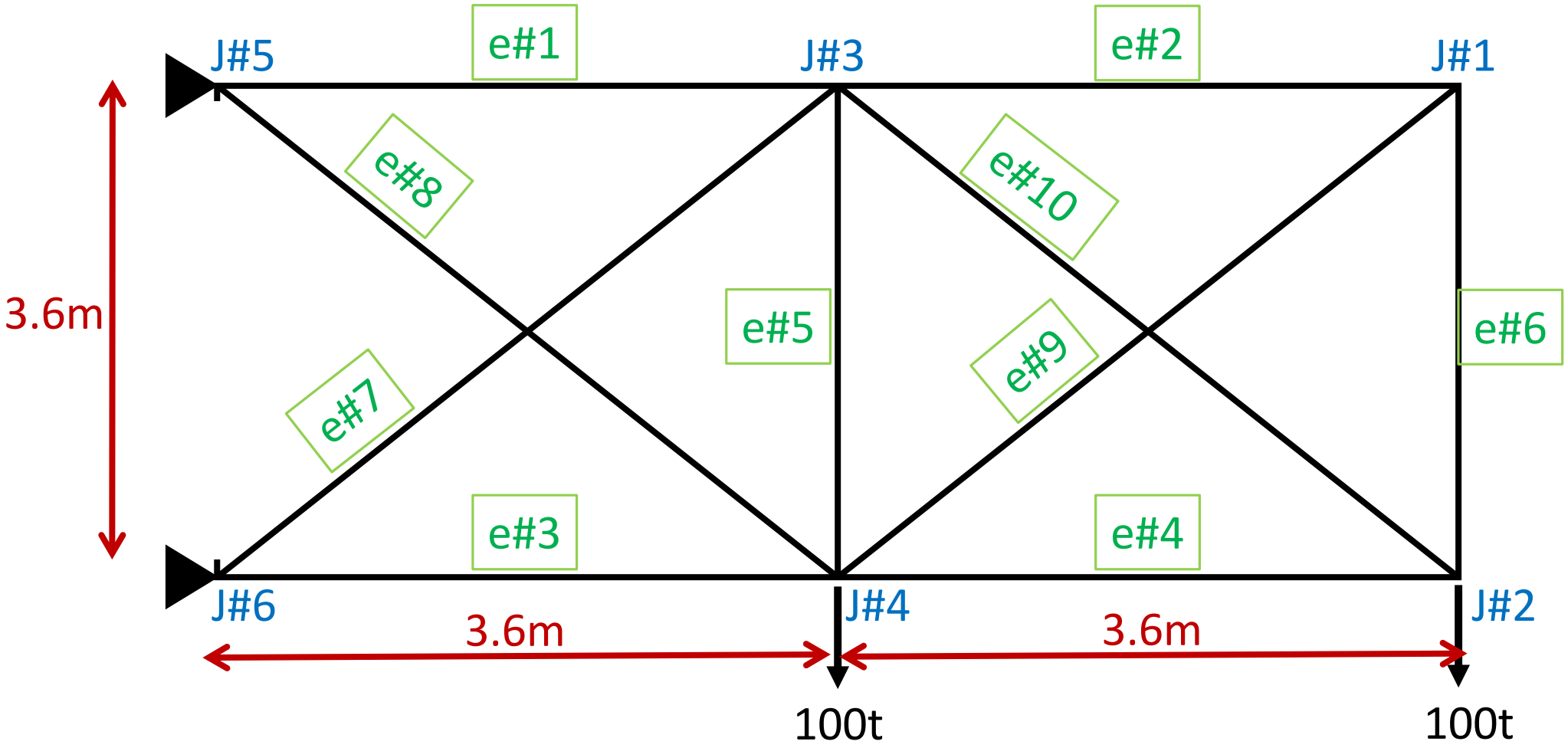
J#5
e#1
J#3
e#2
J#1
e#8
e#10
3.6m
e#5
e#6
e#7
e#9
e#3
e#4
J#6
J#4
J#2
3.6m
3.6m
100t
100t

```
clc;
close all;
clear;

%% Input Geometry && Definitions
% Geometry
Nodes=3.6*[2,1;2,0;1,1;1,0;0,1;0,0];            % Nodes
(m)
Con=[5 3;1 3;6 4;4 2;4 3;2 1;6 3;5 4;4 1;3 2];  %
Connectivity
% Material
E=21000000*ones(1,size(Con,1));                 %
Young's Modulus in t/m2
A=42.6*10^-4*ones(1,10);                        % Area
of each member in m
% Temperature

% Forces
F=zeros(2*size(Nodes,1),1);                     %
Global force vector with setting reactions to zero
F(4,1)=-100;                                    %
Forces in ton
F(8,1)=-100;
% Supports
Sup_DOF = [9 10 11 12];                         %
Global DOFs for supports
Prescribed_BC = [0 0 0 0];                      %
Proscribed value of the displacement

%% Calculations
Tdof=2*size(Nodes,1);                       % Number of
total DOFs
Nelem=size(Con,1);                          % Number of
elements
xi=[Nodes(Con(:,1),1),Nodes(Con(:,1),2)];   % node
Nodesinates i of each element
xj=[Nodes(Con(:,2),1),Nodes(Con(:,2),2)];   % node
Nodesinates j of each element
EA =E.*A;                                   % E*A of
each element
```

```
K = zeros(Tdof);                    % Initialize the global
stiffness matrix
EFT=[2*Con(:,1)-1,2*Con(:,1),2*Con(:,2)-1,2*Con(:,2)];
Ft = zeros(Tdof, 1);                % Initialize the global
thermal load vector
L=zeros(Nelem,1);                   % Intialize length
Vector
c=zeros(Nelem,1);                   % Intialize Cosine
Vector
s=zeros(Nelem,1);                   % intialize sine vector

%% Solver
for i=1:Nelem
%% subroutine to compute the element stiffness matrix
Ke
% Compute the element length L
Xij = xj(i,:) - xi(i,:); % Relative Nodes vector from
node i to node j
e1bar = Xij/norm(Xij);   % Unit vector for the element
axis
c(i) = e1bar(:,1);       % c is cos(phi)
s(i) = e1bar(:,2);       % s is sin(phi)
L(i) = norm(Xij);

% Compute cos(phi) and sin(phi) from the nodal
Nodesinates of node i and j
% Compute the Nodesinate transformation matrix T ( T is
a 4x4 matrix)
T = [ c(i) s(i) 0 0
     -s(i) c(i) 0 0
      0 0 c(i) s(i)
      0 0 -s(i) c(i)];
% Compute the local element stiffness matrix Kebar
Kbar = EA(i)/L(i)*[ 1 0 -1 0
                    0 0 0 0
                   -1 0 1 0
                    0 0 0 0];
% Compute the global element stiffness matrix Ke using
T and Kebar
Ke = T'*Kbar*T;
```

```
%% subroutine to compute the element thermal joint
force Fte
% % Ft_bar = alp(i)*EA(i)*dT(i)*[-1 0 1 0]';
% % % Compute the global thermal joint force vector Ft
using T and Ft_bar
% % Fte = T'*Ft_bar;

%% Assemble the global stiffness K using EFT
K(EFT(i,:), EFT(i,:)) = K(EFT(i,:), EFT(i,:)) + Ke;
% Assemble the global thermal joint force vector Ft
using EFT
Ft(EFT(i,:)) = Ft(EFT(i,:));
end

% Copy K and F+Ft to Khat and Fhat
Khat = K;
Fhat = F+Ft;
% Modify the global force vector using the prescribed
value of displacement
Us = zeros(Tdof,1);
Us(Sup_DOF) = Prescribed_BC;
Fhat = Fhat - K*Us;
% Make rows and columns of Khat related to supports
zero.
Khat(Sup_DOF, :) = 0;
Khat(:, Sup_DOF) = 0;
% Set the corresponding force components to the
prescribed value of the displacement.
Fhat(Sup_DOF) = Prescribed_BC ;
% Put ones to the diagonals related to supports.
Khat(Sup_DOF, Sup_DOF) = eye(length(Sup_DOF));
% Solve the system of the equations and print out the
solution U
U = Khat\Fhat;
disp('U='); disp(U');
% Compute the reaction forces
F = K*U - Ft;
% Print out the reaction F
disp('F='); disp(F');
% Compute the axial force in each element
for i=1:Nelem
```

```
% Extract the external displacement vector Ue from the
global displacement vector
Ue = U(EFT(i,:));
% Compute the axial force
Pe = EA(i)/L(i)*[-c(i) -s(i) c(i) s(i)]*Ue;
% Print out the axial forces
disp(['P',num2str(i),'= ',num2str(Pe)]);
end



Sc=10;
Nodes_Modified=Nodes+Sc*reshape(U,[2,6])';
e=Con(:,1);
f=Con(:,2);

figure()
% axis('square')
hold on
for i=1:length(Con(:,1))
e=Con(i,1);
f=Con(i,2);
X=[Nodes(e,1),Nodes(f,1)];
Y=[Nodes(e,2),Nodes(f,2)];

X2=[Nodes_Modified(e,1),Nodes_Modified(f,1)];
Y2=[Nodes_Modified(e,2),Nodes_Modified(f,2)];

plot(X,Y,'k')
plot(X2,Y2,'--m')

end
hold off
```