

```
%% Plane truss element
```

```
clear all;  
clc;
```

```
%% Geometry
```

```
n_num = input('총 절점의 개수는 몇개입니까?\n>>');
```

```
if n_num/2==0  
    print('절점의 개수는 홀수여야 합니다\n>>');  
end
```

```
n_cd = zeros(n_num, 2);
```

```
n_cd = n_cd + input('각 점의 좌표를 행렬형식으로 입력하시오.\n 예: [node번호 x좌표;] _단위:mm \n>>');
```

```
area = input('단면의 면적을 입력하시오 _단위:mm^2\n>>');
```

```
young = input('탄성계수를 입력하시오 _단위:MPa\n>>');
```

```
element_num = (n_num)/2 - 0.5;
```

```
element_len = zeros(element_num, 2);
```

```
for i=1 : element_num  
    element_len_i = zeros(element_num, 2);  
    length = n_cd(i*2+1, 2) - n_cd(i*2-1, 2);  
    element_len_i(i, 1) = i;  
    element_len_i(i, 2) = length;  
    element_len = element_len + element_len_i;  
end
```

```
end
```

```
element_len
```

```
element_len = 3×2  
    1    1000  
    2    1200  
    3     500
```

```
%% Stiffness matrix
```

```
KG = zeros(n_num, n_num);
```

```
for i=1 : element_num
```

```
    KG_i = zeros(n_num, n_num);
```

```
    element_len_i = element_len(i, 2);
```

```
    KL = young*area/(3*element_len_i)*[7 -8 1; -8 16 -8; 1 -8 7];
```

```
    KG_i((i*2-1):(i*2+1), (i*2-1):(i*2+1)) = KL;
```

```
    KG = KG + KG_i;
```

```
end
```

```
KG
```

```
KG = 7×7  
103 ×  
    0.2800    -0.3200     0.0400         0         0         0         0  
   -0.3200     0.6400    -0.3200         0         0         0         0
```

0.0400	-0.3200	0.5133	-0.2667	0.0333	0	0
0	0	-0.2667	0.5333	-0.2667	0	0
0	0	0.0333	-0.2667	0.7933	-0.6400	0.0800
0	0	0	0	-0.6400	1.2800	-0.6400
0	0	0	0	0.0800	-0.6400	0.5600

```

%% Boundary condition
BC = zeros(n_num,2);
BC = BC + input('각 점의 경계조건을 행렬형식으로 입력하세요 fixed :1, free:0');

LD = zeros(n_num,2);
LD= LD + input('각 점의 하중조건을 입력하세요. \n 예:[node번호 하중] _단위:N');

cnt = 0;
KG_p = KG;
for i=1:n_num
    chk = BC(i,2);
    if chk == 1
        KG_p (i-cnt,:)=[];
        KG_p(:,i-cnt)=[];
        cnt=cnt+1;
    end
end

cnt1=0;
LD_p = LD(:,2);
for i=1:n_num
    chk1 = BC(i,2);
    if chk1 == 1
        LD_p(i-cnt1,:)=[];
        cnt1=cnt1+1;
    end
end

d_p = inv(KG_p)*LD_p;

d=zeros(n_num,1);
for i =1:n_num
    d_i=zeros(n_num,1);
    chk = BC(i,2);
    if chk == 0
        d_i(i)=d_p(1);
        d_p(1)=[];
    end
    d=d+d_i;
end

d

```

```

d = 7×1
    0
    0.2025
    0.0926
   -0.0706

```

-0.0463
-0.0231
0

F = KG *d

F = 7×1
-61.1111
100.0000
-0.0000
-50.0000
0.0000
0
11.1111