Consider the truss structure shown above. Truss is composed of equal length members forming 2 equilateral triangles. Each node has two degrees of freedom, translational displacements in the x and y directions respectively. Obtain the structural stiffness matrix \([K]\), and the load vector \([R]\) for the given structure utilizing the algorithm given in Figure 2.7-4 on page 44.

**Given information:**

- A: 1 cm\(^2\) for each member
- E: 70 GPa for each member
- \(\alpha\): 23x10\(^{-6}\) for each member
- W: 0.081 kg for each member
- F: 1000 N
- g: 9.81 m/s\(^2\)
- T: 100 °C for each element (temperature increase)
**Notes:** The following notes are just guidelines. You can design your code in any way you wish. However, you must make sure that your code is general and can easily be adapted to other structures like 3D truss or frame etc. For instance, if you have 4 elements in your structure, you should not use #4 within your code, instead you should assign a variable name to it like NUMEL and use it.

- Initially, you should read in the total number of elements : NUMEL from a data file
- Next, you should read in the following values from a data file in a loop (1 to total number of elements) in the following format for each of the element of the truss structure
  - Structural node numbers: elements of the array NOD(I,N); NOD(1,N), NOD(2,N),
  - X and Y coordinates of the nodes of each element: X(1), Y(1), X(2) and Y(2),
  - Cross sectional area of the element (A(N)),
  - Young’s modulus of the element (E(N)),
  - Coefficient of thermal expansion of the element (α(N)),
  - Temperature change of the element (T(N)),
  - Weight of the element (W(N))
- In this loop, at each step you should calculate:
  - Length of the elements (L(N))
  - Angular orientation of each element (β(N))
- Then, you should calculate the element stiffness matrices \([k]\), and element load vectors \(\{r_e\}\) within the subroutine ELEMNT. Subroutine ELEMNT is assumed to return element matrices \([k]\) and \(\{r_e\}\) in arrays SE(I,J) and RE(I).

Note that you should transfer the information such as angles, lengths, areas etc. to the subroutine via COMMON blocks or arguments of the CALL statement.

- Output of the program should give the assembled structural stiffness matrix \([K]\) in array S(K,L) in matrix form and the assembled structural load vector \(\{R\}\) in array R(K) in vector form.
- The external load vector \(\{P\}\) should be added to the load vector outside the main loop.
The program should be complete in itself utilizing the algorithm given in Figure 2.7-4 on page 44. **Use appropriate comment statements within the code to explain what is done at each step. If you do not explain your steps with comment statements you will lose points.**