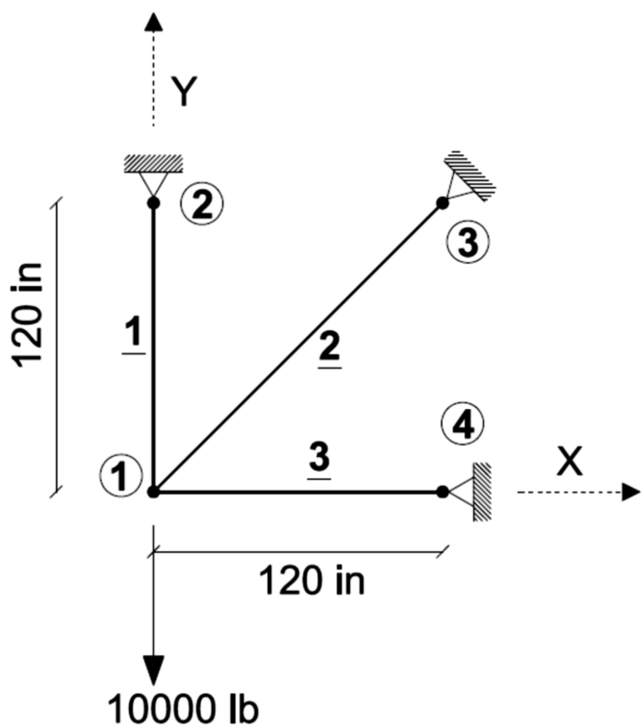
M1. (MATLAB Assignment)

In this section, you will accomplish the simple MATLAB program you have developed in the previous assignments. In this homework, you will

- Implement an assembly approach to assemble the system global stiffness matrix
- Form the global nodal load vector
- Impose boundary conditions
- Solve KU = F to obtain nodal displacements

Download the skeleton file **hw6.m** and **input.txt** before you begin. The input file includes the data (including geometry, boundary conditions, material properties, and loading) required to define the same planar truss given in the previous assignments with material properties $E = 30 \times 10^6\ psi$ and $A = 2\ in^2$ for all elements, and a downward concentrated load applied at node 1.



The skeleton m-file reads the input file and generates the required variables including:

- `nNode` : Number of nodes,
- `nElem` : Number of elements,
- `coordinates`: An nNode-by-2 matrix including x and y coordinates,
- `elements` : An nElem-by-2 matrix including the element connectivity data,
- E : An nNode-by-1 matrix includes the Young's modulus of elasticity values,
- A : An nNode-by-1 matrix includes the cross-sectional area of each element,
- rx: An nNode-by-1 matrix indicates whether a node is free (=1) or restrained (=0) in x-direction,
- ry: An nNode-by-1 matrix indicates whether a node is free (=1) or restrained (=0) in y-direction.
- fx: An nNode-by-1 matrix includes the nodal forces component in the x-direction, and
- fy: An nNode-by-1 matrix includes the nodal forces component in the y-direction.

Your task:

To accomplish this assignment, you need to complete the following tasks:

a. Initialize variable K with zeros to store the assembled stiffness matrix in the next step. The size of this matrix is a function of "number of nodes" and "degrees of freedom per each node". These parameters are defined in lines 36 and 66.
   - Hint: use MATLAB function "zeros" which creates a matrix of all zeros

b. Assemble total stiffness matrix. The element stiffness matrices (in global coordinates) are calculated in line 102. You need to find a way to add each element stiffness matrix to its corresponding place in the total stiffness matrix K.
   - Hint: An elegant approach for doing that in MATLAB would be finding the indices of corresponding elements in the total stiffness matrix and using matrix computation features in MATLAB to add the element stiffness matrix values to the total stiffness matrix, in the same for-loop i.e. `K(index,index) = K(index,index) + k` (See line 119)

c. Form the total nodal force vector. The matrix of Nodal forces f is defined in line 125. Use this matrix to define vector F with the following form:

$$F = \begin{Bmatrix} Fx_1 \\ Fy_1 \\ Fx_2 \\ Fy_2 \\ \vdots \\ Fx_n \\ Fy_n \end{Bmatrix}$$

   - Hint: You may consider using either function "reshape" or for-loops

d. Solve the system of equations (KU = F) to obtain U and briefly explain why you get that result. Write your answer in a comment line (start with %) in the same m-file in less than 20 words.
   - Hint: In general, the solution of the system of equations $Ax = b$, is $x = A^{-1}b$. In MATLAB, you can use x = inv(A)*b or simply x = A\b.

e. Now, impose the boundary conditions. Boundary conditions are defined by matrix bc (line 142). In this matrix, 0 indicates a restrained degree of freedom and 1 indicates a free (unrestrained) degree of freedom. You can use this binary matrix to find a way to enforce boundary conditions.

f. Now, solve the system of equations after imposing boundary conditions and obtain the nodal displacements.

Important Note:

- You are welcome to implement any other approach for conducting the assembly procedure, imposing boundary conditions, and solving the system of equations.
- You are not allowed to hardcode any values in your code. **You will not get any credit if hardcode any length, stiffness matrix, or transformation matrix**. Your program must be able to handle any other input file.

Submission:

Upload your completed version of **hw6.m** to Canvas and **attach** a **hard copy** of (1) <u>your code</u> and (2) <u>Command Window outputs</u> to your submission in class.