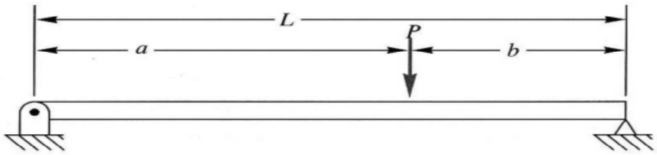
Consider a simply supported beam to which a single concentrated load is applied:



Given: *P* is the load, *E* is the modulus of elasticity, *I* is the moment of inertia, *a* is the distance from the left end of the beam to the load, and $b = L - a$ is the distance from the right end of the beam.

For *a* ≤ *b*, the maximum deflection is calculated as:

$$Max\_Deflection = \frac{-Pa(L^2 - a^2)^{3/2}}{9\sqrt{3}EIL}$$

the deflection at the load is calculated as:

$$Deflection\_at\_Load = \frac{-Pa^2b^2}{3EIL}$$

and the deflection at the beam's center is calculated as:

$$Deflection\_at\_Center = \frac{-Pa(3L^2 - 4a^2)}{48EI}$$

For *a* > *b*, simply interchange *a* and *b* in the preceding equations.

Write a main program that:

- Inputs values for *L*, *P*, *E*, *I*. and an *increment* by which to move the load
- Outputs headings for a 4-column table whose contents are described in (3) below
- Uses a for loop to:
  - (1) Move the load position along the beam from the left end to the right end in the specified increment
  - (2) At each position, `main()` must call a void function that receives L, P, E, I, and the current load position and that calculates and passes back the 3 deflections to `main()`
  - (3) `main()` then outputs a row of values — the load position and the three deflections — into the table.

Use the `setw()` *I/O-manipulator to set the column widths so that nice even columns are produced in the table, aligned under the column headings***.** Execute your program twice:

(1) With $L = 360$ inches, $P = 24{,}000$ pounds, $E = 30 \times 10^6$ psi, $I = 795.5\text{in}^4$, 6 inch increment

(2) With your own set of inputs.

Beam Project Grade Sheet

Name: ____

Hand in:

This grade sheet attached to:

- A printout of your program (all source code if you use a library)
- Two execution traces — one with the inputs given in the project description and one with your own inputs.

Category	**Points Possible**	**Points Received**
Correctness of deflection-calculator function	30	_______
Correctness of main program (including following instructions)	30	_______
Output/Build Window; Program Executions; Format of Tables Output	15	_______
Structure & Organization (efficiency, no unnecessary code, arrangement of statements, doesn't waste memory)	5	_______
Documentation (opening doc. and doc. for the function)	10	_______
Style & Readability (meaningful names, indentation, alignment, white space (between lines and within lines)	10	_______
TOTAL	**100**	_______