Homework Assignment 007 (2020)

Simply Supported Beam with Two Concentrated Loads

Create a Win application project named as <yourStudentID>SimplySupportedBeamAss007. Remember to rename your Form1.cs file to have a meaningful class name.

Add a class (namely, e.g., SimplySupportedBeamWithTwoConcentratedLoads) to represent a simply supported beam with two concentrated loads. Your class should define the related data fields with initial values assigned: e.g., float **`L, a1, a2, P1`**, and **`P2`**. They are declared **`private`** (by default) for data encapsulation. However for easier access from your UI, you will define their corresponding public properties. Property values can be modified by the user via proper UI controls. Therefore, in your user interface form, provide necessary controls to host these values for value updating. Exercise creating definitions of properties for these member data and using property grid to host these properties. In setter of each property, implementing error-proof value setting.

This class will provide servicing methods (public functions) that returns shear force, bending moment, and even deviation (deflection) for a given location on the beam. In this assignment, at least complete methods

`public float getShearForce( float x )`, and

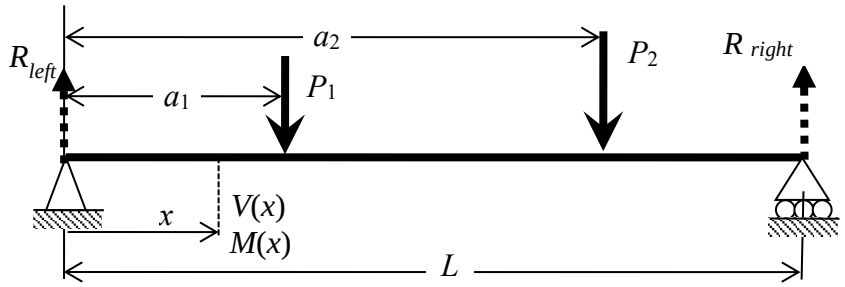
`public float getBendingMoment( float x )`, that calculate and return the shear force and bending moment at position **`x`** of the beam, respectively. Equations for moment and force computations are shown bellows.

$$R_{left} = \left(P_1\left(L-a_1\right)+P_2\left(L-a_2\right)\right)/L$$

$$R_{right} = \left(P_1a_1+P_2a_2\right)/L$$

$$V(x)=\begin{cases} R_{left}, \text{if } 0 \le x \le a_1 \\ R_{left}-P_1, \text{else if } a_1 \le x \le a_2 \\ -R_{right}, \text{else if } a_2 \le x \le L \end{cases}$$

$$M(x)=\begin{cases} R_{left}x, \text{if } 0 \le x \le a_1 \\ R_{left}x-P_1\left(x-a_1\right), \text{else if } a_1 \le x \le a_2 \\ R_{right}\left(L-x\right), \text{else if } a_2 \le x \le L \end{cases}$$



Note that since the class has already defined the related data fields (data members of this class such as **`float L, a1, a2, P1,`** and **`P2`**) the two member functions can access these data members directly. Usually, public properties are declared for accessing from outside, while private data fields are accessed directly within the class.

```
public class SimplySupportedBeamWithTwoConcentratedLoads
{
    float l = 100.0f;
    public float L
    { get{ return l; } set { l = value; } }

    public float a1 = 30.0f, a2 = 70.0f;
    public float p1 = 50.0f, p2 = 150.0f;

    public float getShearForce( float x )
    {
        float Rleft, Rright, V = 0.0f;
        // ??????
        return V;
    }

    public float getBendingMoment( float x )
    {
        float Rleft, Rright, M = 0.0f;
        // ??????
        return M;
    }
}
```

In your form, lay out proper UI controls to host the attributes of the beam (L, a1, a2, P1, and P2). In addition, prepare a **`ListBox`**(or **`RichTextBox`**) to list the location, computed shear force, and moment on the beam starting from **`x = 0`** up to **`x = L`** with an increment delta x **`= 0.5`** along the beam. In your form class, you need to define an object of the beam class and instantiate it (new it) right after the initialization function call, **`InitializeComponent( )`**, within the constructor of the form. You will use this object (name it by yourself) to access its member functions and change its properties directly.

You need to use **`for`**-statement to loop through each **`x`** value and call the provided methods to get shear force and bending moment for display. The property of the **`ListBox`** that hosts the

listed items is **`Items`**. Make sure your clear the list (call method ListBoxName.**`Items.Clear()`**) first, before adding any item to it. To add an item to the list is simply calling method ListBoxName.**`Items.Add(`** yourItem **`)`**;. Note that the item to be added can be anything (type is **`object`**)

The following is a snapshot of a sample assignment for your reference.