

```
In [11]: import numpy as np
import pandas as pd
import IPython.display as ipd
import sympy as sy

x, y, z = sy.symbols('x y z')
sy.init_printing()
```

## LISTA 5 - QUESTÃO 1

```
In [12]: ipd.Image(filename='L5Q1.png')
```

Out[12]: **1ª Questão)** Localize o centroide  $(\bar{x}, \bar{y})$  da área da seção transversal da calha mostrada na Figura 1.

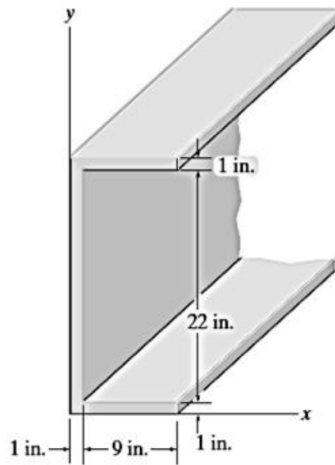


Figura 1.

### Objetivos:

Calcular o centroide da peça mostrada

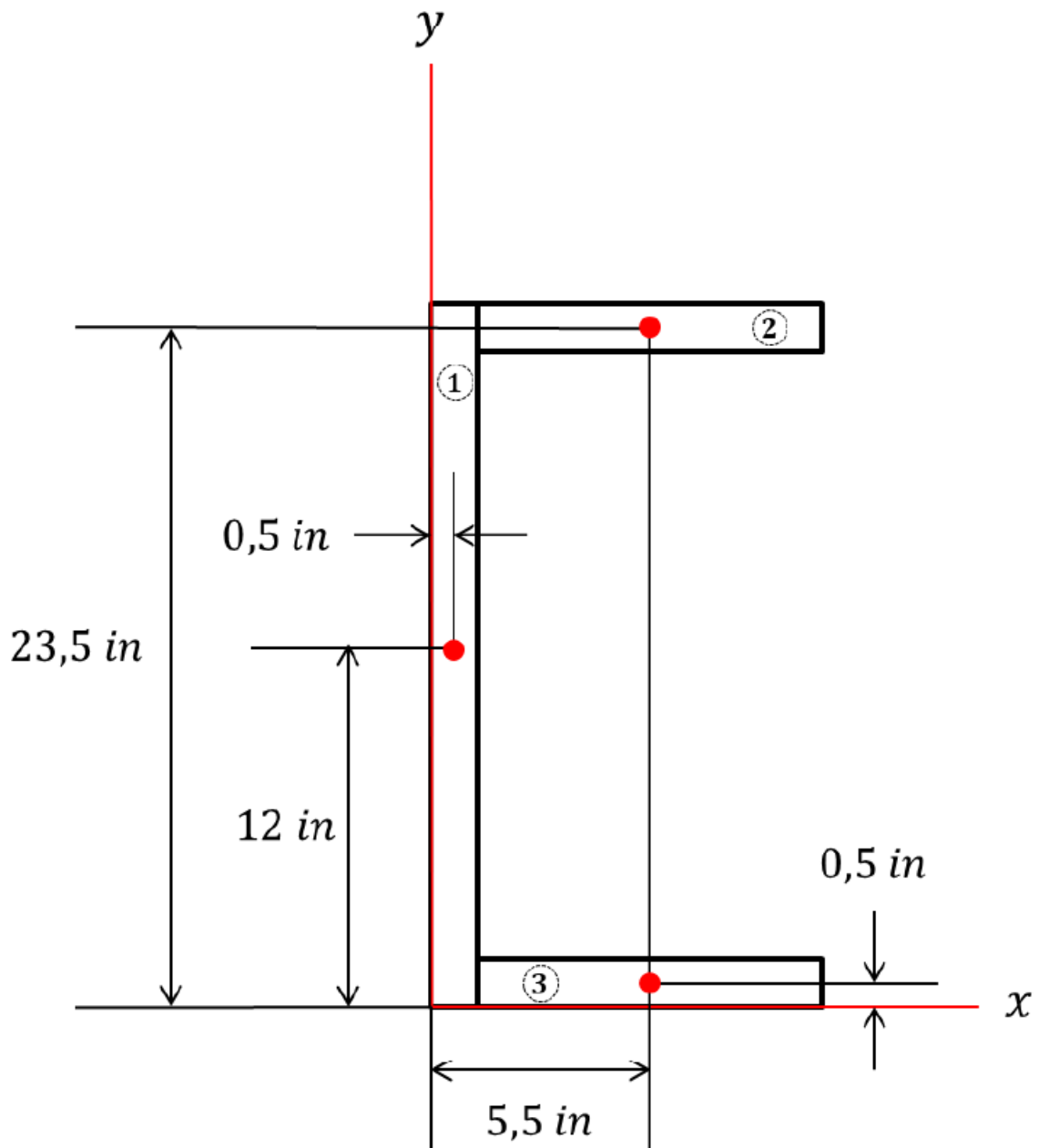
Dados a seguir

```
In [13]: espessura = 1 # [in]
Ly = 22 # [in]
Lx = 9 # [in]
```

A peça possui três áreas a serem trabalhadas. podemos dividir em três áreas, 1, 2 e 3. vamos organizar na forma de vetores posição, o centroide de cada uma dessas áreas:

```
In [14]: ipd.Image(filename='L5Q1-2.png')
```

Out[14]:



Calculando os centroides e organizando uma tabela (data\_centroides) para acessar os dados depois:

```
In [15]:
centroide_A2 = [Lx/2 + espessura, Ly + espessura*3/2]
centroide_A1 = [espessura/2, (2*espessura+Ly)/2]
centroide_A3 = [Lx/2 + espessura, + espessura/2]

data_centroides = pd.DataFrame({
    'Eixos':[x, y],
    'Centroide A1':centroide_A1,
    'Centroide A2':centroide_A2,
    'Centroide A3':centroide_A3
})
data_centroides = data_centroides.T
data_centroides
```

```
Out[15]:
```

	0	1
<b>Eixos</b>	x	y
<b>Centroide A1</b>	0.5	12.0
<b>Centroide A2</b>	5.5	23.5
<b>Centroide A3</b>	5.5	0.5

Tabelando os valores das áreas

```
In [16]: A1 = espessura * (Ly + 2 * espessura)
A2 = espessura * Lx
A3 = A2
data_areas = pd.Series({
    'Área A1':A1,
    'Área A2':A2,
    'Área A3':A3
})
data_areas
```

```
Out[16]: Área A1    24
Área A2     9
Área A3     9
dtype: int64
```

Esse bloco abaixo é só pra mudar o "tipo" de dados para jogar dentro da função dot (produto interno)

```
In [17]: Y = data_centroides.iloc[1:,1:2].squeeze()
X = data_centroides.iloc[1:,0:1].squeeze()
areas = data_areas.squeeze()
```

aqui foi realizado o produto interno das coordenadas X dos centroides com os valores das áreas e dividido pela soma (sum(areas)) das áreas analisadas:

```
In [ ]: centroide_compostoX = np.dot(X, areas)/sum(areas)
centroide_compostoX
```

Fazendo o mesmo para Y, temos:

```
In [ ]: centroide_compostoY = np.dot(Y, areas)/sum(areas)
centroide_compostoY
```

```
In [ ]: print('As coordenadas do centroide da área composta são X: {:.3f} in e Y: {:.3f} in')
```