A

The beam will experience the highest moment of x on the point furthest away from the wall because it is an extremity and it doesn't have a wall creating a force in the opposite direction in this location

B

Given that the beam is symmetric along the y axis, the center x-coordinate will be at $\frac{b}{2}$

$$y_c = \frac{\sum yA}{\sum A}$$
$$= \frac{2\frac{b}{2}(bt)+t(2t(b-2t))}{2bt+2t(b-2t)}$$

Since t^n 0 **when** $n > 1$ **,** $t(2t(b-2t)) = 0$ **and** $4t^2 = 0$

$$y_c = \frac{b^2t}{2bt+2tb}$$
$$= \frac{b^2t}{4tb} = \frac{b}{4}$$

This has been proven with the following code:

In [183]:

```
import numpy as np
import sympy as spy

b, t, M_x, y = spy.symbols('b t M_x y')

left_arm = {'x': t, 'y': b, 'x_c': t/2, 'y_c': b/2}
center_beam = {'x': b-2*t, 'y': 2*t, 'x_c': b/2, 'y_c': t}
right_arm = {'x': t, 'y': b, 'x_c': b-(t/2), 'y_c': b/2}

y_c = (left_arm['y_c'] * left_arm['x'] * left_arm['y'] + center_beam['y_c'] * center_beam['x'] * center_beam['y'] + right_arm['y_c'] * right_arm['x'] * right_arm['y'])/(left_arm['x'] * left_arm['y'] + center_beam['x'] * center_beam['y'] + right_arm['x'] * right_arm['y'])
x_c = (left_arm['x_c'] * left_arm['x'] * left_arm['y'] + center_beam['x_c'] * center_beam['x'] * center_beam['y'] + right_arm['x_c'] * right_arm['x'] * right_arm['y'])/(left_arm['x'] * left_arm['y'] + center_beam['x'] * center_beam['y'] + right_arm['x'] * right_arm['y'])

x_c, y_c = spy.expand(x_c), spy.expand(y_c)
for m in range(2, 5):
    x_c = x_c.subs(t**m, 0)
    y_c = y_c.subs(t**m, 0)

x_c, y_c = x_c.subs(t**2, 0), y_c.subs(t**2, 0)
print(f'x_c={x_c}' , f'y_c={y_c}')
```

```
x_c=b/2 y_c=b/4
```

C

In [184]:

```
I_xx_1 = (1/12) * right_arm['x'] * right_arm['y']**3 + right_arm['x'] * right_arm['y'] * (right_arm['y_c'] - y_c)**2
I_xx_2 = (1/12) * center_beam['x'] * center_beam['y']**3 + center_beam['x'] * center_beam['y'] * (center_beam['y_c'] - y_c)**2
I_xx_3 = (1/12) * left_arm['x'] * left_arm['y']**3 + left_arm['x'] * left_arm['y'] * (left_arm['y_c'] - y_c)**2
I_xx = I_xx_1 + I_xx_2 + I_xx_3
I_yy_1 = (1/12) * right_arm['y'] * right_arm['x']**3 + right_arm['y'] * right_arm['x'] * (right_arm['x_c'] - x_c)**2
I_yy_2 = (1/12) * center_beam['y'] * center_beam['x']**3 + center_beam['y'] * center_beam['x'] * (center_beam['x_c'] - x_c)**2
I_yy_3 = (1/12) * left_arm['y'] * left_arm['x']**3 + left_arm['y'] * left_arm['x'] * (left_arm['x_c'] - x_c)**2
I_yy = I_yy_1 + I_yy_2 + I_yy_3

I_xy_1 = right_arm['x'] * right_arm['y'] * (right_arm['y_c'] - y_c) * (right_arm['x_c'] - x_c)
I_xy_2 = center_beam['x'] * center_beam['y'] * (center_beam['y_c'] - y_c) * (center_beam['x_c'] - x_c)
I_xy_3 = left_arm['x'] * left_arm['y'] * (left_arm['y_c'] - y_c) * (left_arm['x_c'] - x_c)
I_xy = I_xy_1 + I_xy_2 + I_xy_3

I_xx, I_yy, I_xy = spy.expand(I_xx), spy.expand(I_yy), spy.expand(I_xy)
for m in range(2, 5):
    I_xx = I_xx.subs(t**m, 0)
    I_yy = I_yy.subs(t**m, 0)
    I_xy = I_xy.subs(t**m, 0)

print(f'I_xx={I_xx}' , f'I_yy={I_yy}', f'I_xy={I_xy}')
```

```
I_xx=0.416666666666667*b**3*t I_yy=0.666666666666667*b**3*t I_xy=0
```

In [185]:

```
sigma_z = M_x/I_xx * y
sigma_z
```

Out[185]:

$$\frac{2.4 M_x y}{b^3 t}$$

D

The points at which the stress whill be the largest are the ones with the highest y, regardless of their x positions. This means that both top corners will experience the same stress along the corners further away from the wall