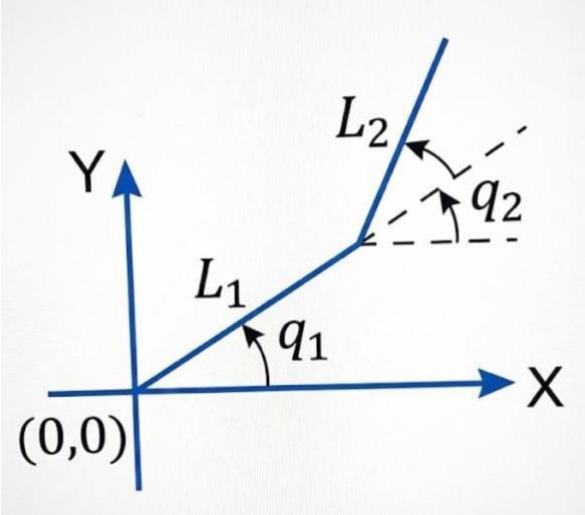
Task 1:

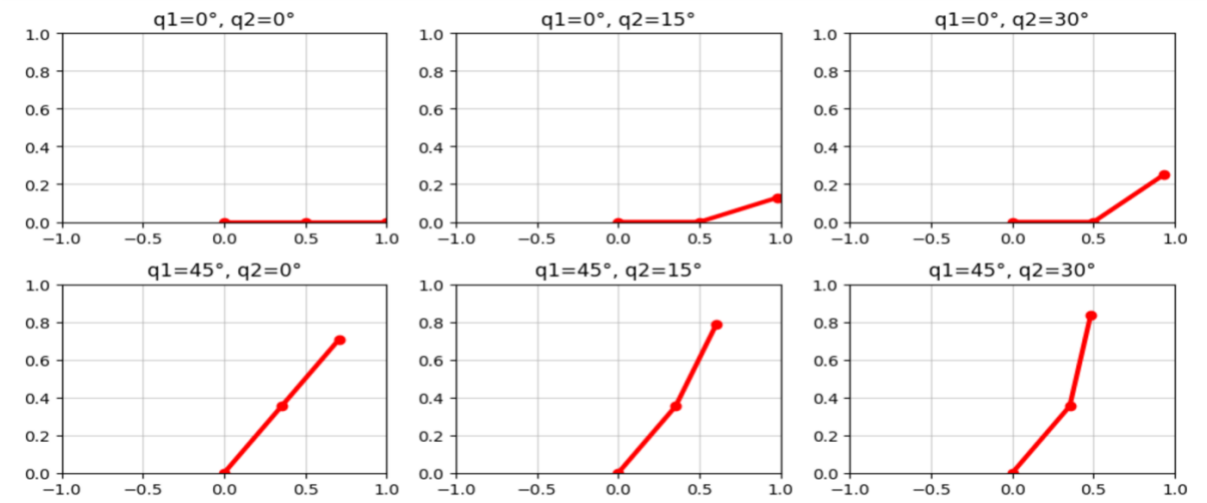
Standard configuration for the robotic arm.

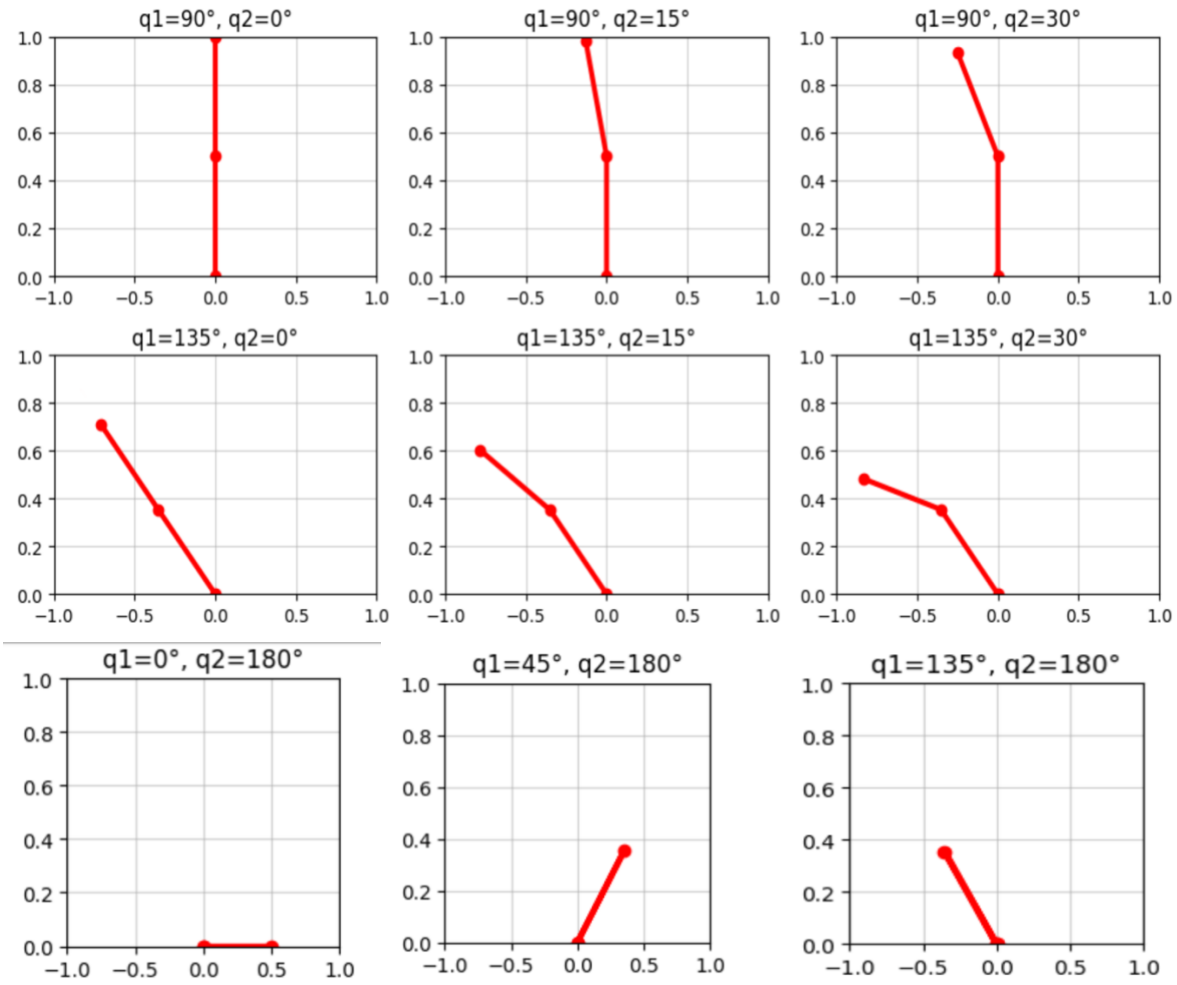


Co-ordinate for the elbow joint: (L1cos(q1), L1 sin(q1))
Co-ordinate for the end-effector: (L1cos(q1) +L2cos(q1+q2), L1sin(q1) +L2sin(q1+q2))
The co-ordinate for elbow joint stays the same in every case.
For a straight arm: q2 would be 0, the end effector co-ordinate would change to ((L1+L2) cos(q1), (L1+L2) sin(q1).
For a bent arm, the original expression stays the same.
For a folded arm q2 would be -180. So ((L1cos(q1)- L2cos(q1), L1sin(q1) – L2sin(q1)).
As given, we can substitute L1=L2= l and get final set of expressions.

Task 2:

We have already discussed the co-ordinates of the elbow joint and end effector above. For the visualization, as asked we have taken L1 and L2 as a common fixed quantity. And plotted the robotic arm using two links, the first one for elbow joint and base and the second one for elbow joint and end effector and varied the values for the angles q1 and q2. On increasing q1, the first link seems to be rotating anti-clockwise around the base point. On increasing q2, the second link seems to be curling up, relative to the first link.






```python
import numpy as np
import matplotlib.pyplot as plt

def roboticarm(l1, l2, q1, q2):
    x0, y0 = 0, 0
    x1 = l1 * np.cos(np.radians(q1))
    y1 = l1 * np.sin(np.radians(q1))
    x2 = x1 + l2 * np.cos(np.radians(q1 + q2))
    y2 = y1 + l2 * np.sin(np.radians(q1 + q2))
    return [x0, x1, x2], [y0, y1, y2]

l1, l2 = 0.5, 0.5
q1_vals = [0, 45, 90, 135]
q2_vals = [0, 15, 30,180]

a = len(q1_vals)
b = len(q2_vals)
plt.figure(figsize=(10, 10))
plot_num = 1
for q1 in q1_vals:
    for q2 in q2_vals:
        plt.subplot(a,b, plot_num)
        x, y = roboticarm(l1, l2, q1, q2)
        plt.plot(x, y,'ro-', linewidth=3)
        plt.title(f"q1={q1}°, q2={q2}°")
        plt.xlim(-1, 1)
        plt.ylim(0, 1)
        plt.grid(True, alpha=0.6)
        plot_num += 1
plt.tight_layout()
plt.show()
```