Given the lengths the links:
To the elbow: $l1=1$ unit
To the end-effector: $l2=1$ unit
Task 1: Calculating positions

The elbow is of length l1 and at an angle q1

Position of the elbow joint from the base(0,0):

$xe = l1cos(q1)$
$ye = l1sin(q1)$

The end-effector is at a distance of l2 from elbow joint and at an angle q2 relative to the elbow joint. Hence it is at a total angle of q1+q2

Position of the end-effector from the base (0,0):

$x = xe + l2cos(q1+q2)$
$y = ye + l2sin(q1+q2)$

Taking three configurations:

1. Straight Arm Configuration
2. Bent Elbow Configuration
3. Folded Arm Configuration

1. Straight Arm Configuration:
   Since the arm is straight, this implies that the relative angle between the elbow joint and end-effector will be 0, i.e q2=0
   q1 can be anything, as it is basically it's orientation in space.
   Taking q1=0:
   $xe = l1 = 1$ unit
   $ye = 0$
   $x = xe + l2 = 1 + 1 = 2$ units
   $y = ye + 0 = 0 + 0 = 0$

2. Bent Elbow Configuration:
   Since the arm is bent, then, q2 can be anything except 0 (as it will make the arm straight) and +- pi (as it will fold it).
   Again q1 can be anything as it is just it's orientation in space.
   Taking q1=q2=pi/4:
   xe = l1cos(pi/4) = 1/root 2
   ye = l1sin(pi/4) = 1/root 2
   x = xe + l2*0 = 1/root 2
   y = ye + l2*1 = 1/root 2 + 1

3. Folded Arm Configuration:
   Since the arm is folded, then it implies that q2 will be either pi or -pi.
   Again q1 can be anything.
   Taking q1 = pi/4 and q2 = -pi
   xe = l1cos(pi/4) = 1/root 2
   ye = l1sin(pi/4) = 1/root 2

x = xe + l2cos(-3pi/4) = 0
y = ye + l2sin(-3pi/4) = 0

Task 3: Explanations

q1 is the elbow joint angle, and increasing this angle will lift the arm, whereas decreasing the angle will lower the arm, in short, it will change the orientation of the workspace.

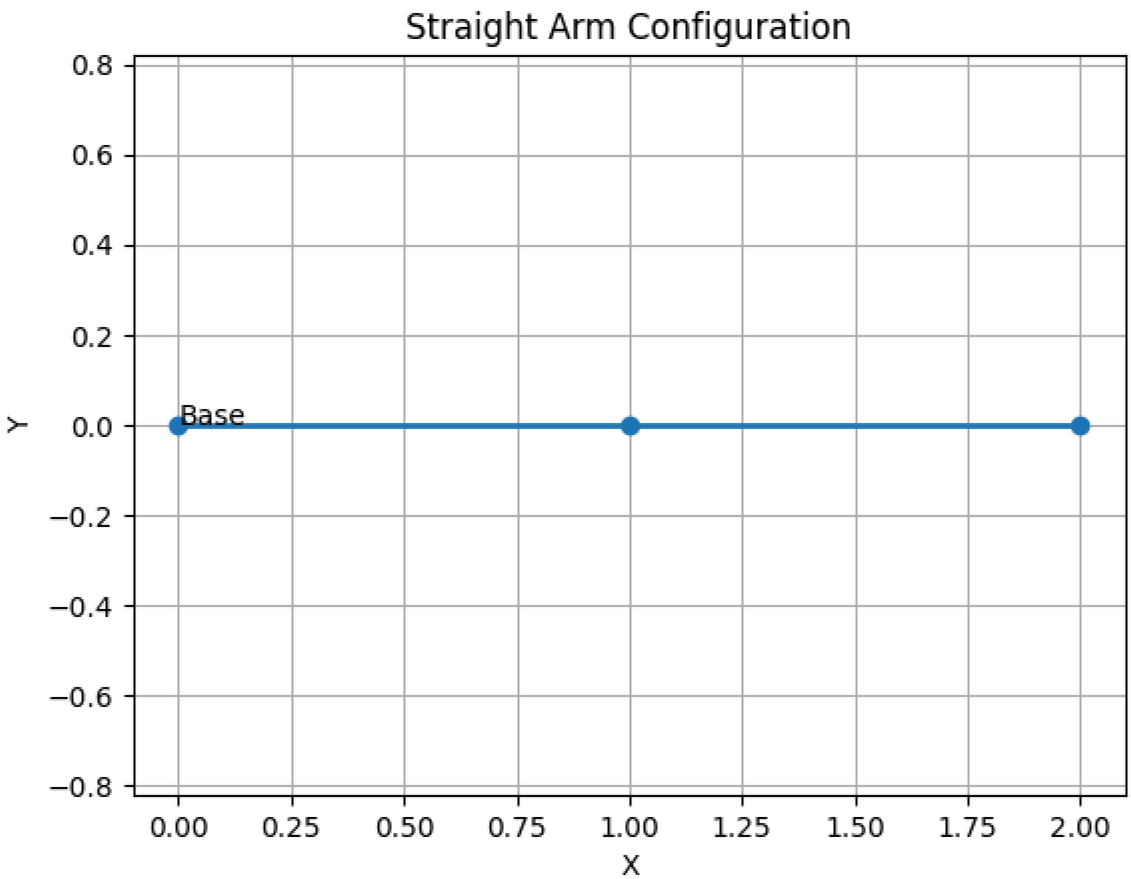
q2 is the relative angle made between the two links, so, increasing/decreasing this angle will bend the arm more and when it is equal to 180 deg, it will fold the arm.

The workspace that the arm can reach to, is the annulus having radius lying between |l1-l2|(where it is a folded arm) and l1+l2(where it is a straight arm). Given, l1=l2=1, the arm can reach to any point lying between radius 0 and 2.

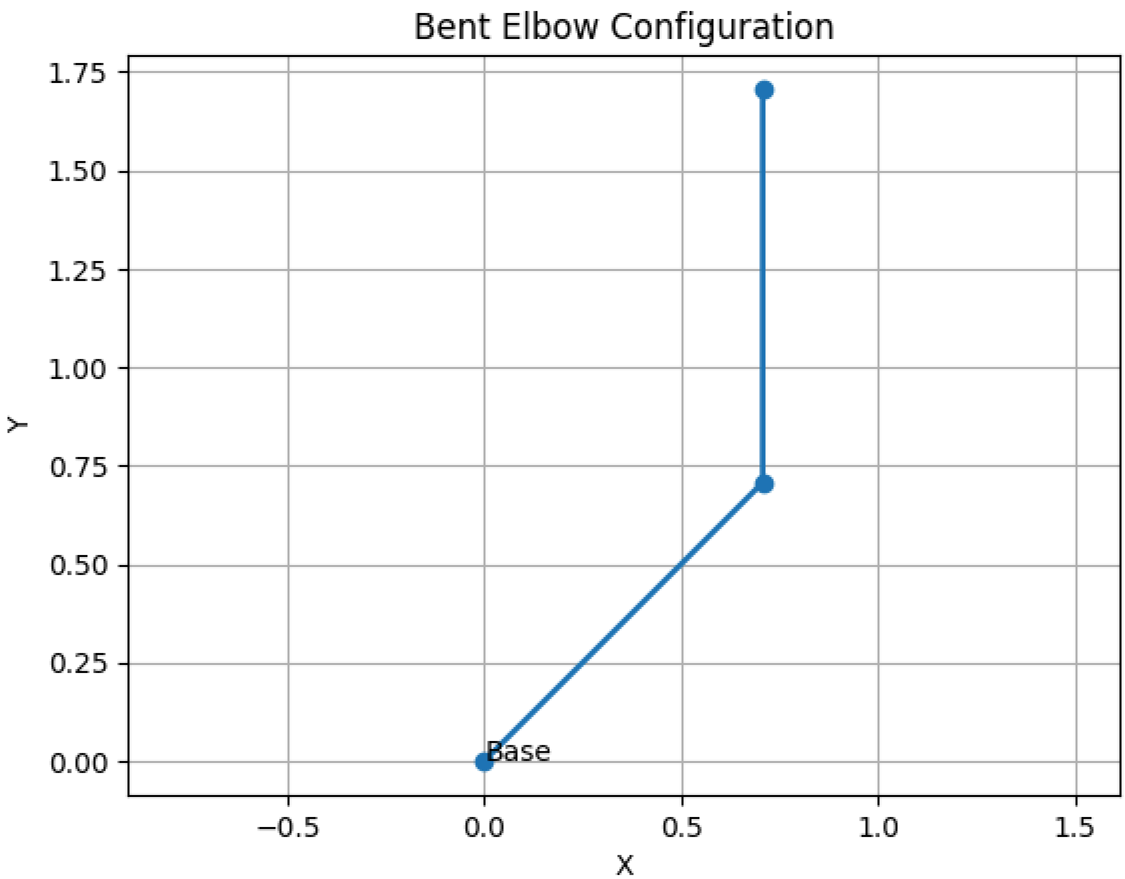
Task 2: Python script and images, shown below:

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt

# Given link lengths
l1 = 1.0
l2 = 1.0
def forward_kinematics(q1, q2):
    # Elbow position
    xe = l1 * np.cos(q1)
    ye = l1 * np.sin(q1)

    # End-effector position
    x = xe + l2 * np.cos(q1 + q2)
    y = ye + l2 * np.sin(q1 + q2)

    return (xe, ye), (x, y)
def plot_arm(q1, q2, title):
    elbow, ee = forward_kinematics(q1, q2)
    x_points = [0, elbow[0], ee[0]]
    y_points = [0, elbow[1], ee[1]]

    plt.figure()
    plt.plot(x_points, y_points, '-o', linewidth=2)
    plt.text(0, 0, "Base")
    plt.axis('equal')
    plt.grid(True)
    plt.xlabel("X")
    plt.ylabel("Y")
    plt.title(title)
    plt.show()
```

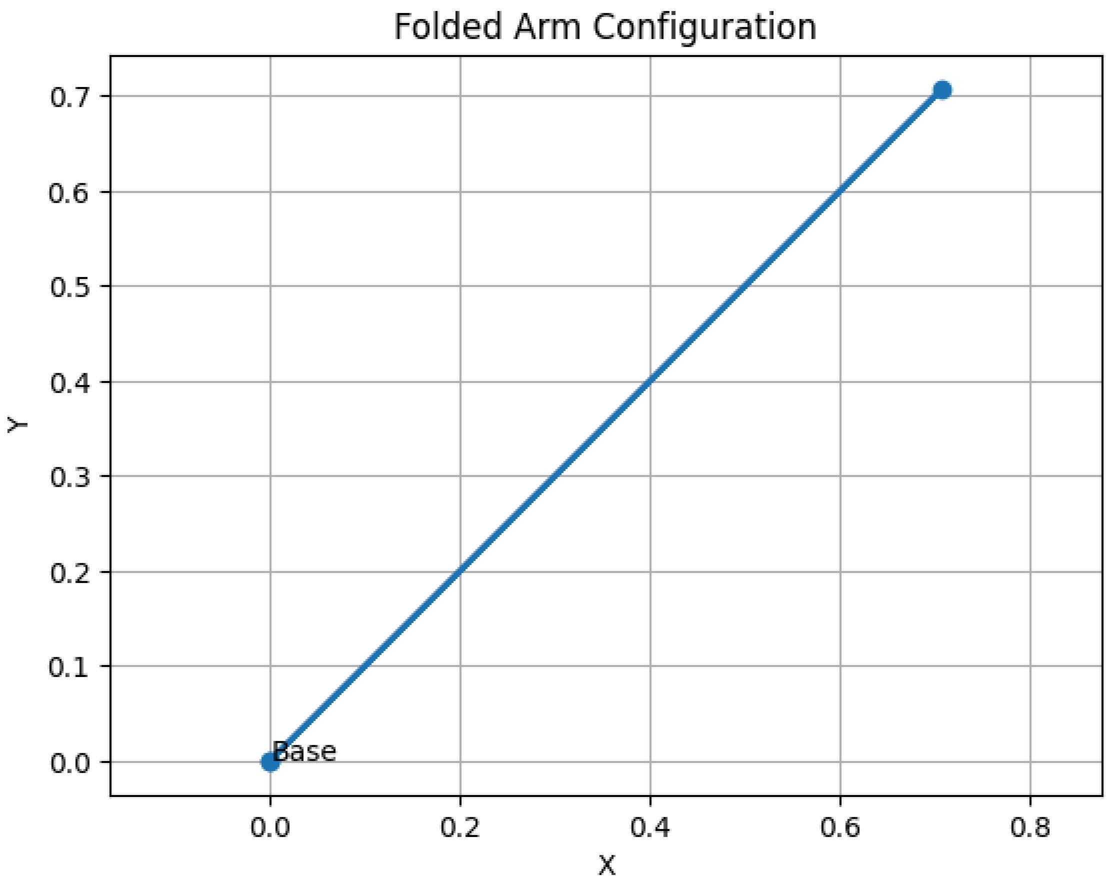
In [2]:
```
q1 = 0
q2 = 0
plot_arm(q1, q2, "Straight Arm Configuration")
```




In [3]:
```
q1 = np.pi / 4
q2 = np.pi / 4
plot_arm(q1, q2, "Bent Elbow Configuration")
```




In [4]:
```
q1 = np.pi / 4
q2 = -np.pi
plot_arm(q1, q2, "Folded Arm Configuration")
```


Folded Arm Configuration
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Y
Base
0.0
0.2
0.4
0.6
0.8
X