Followings are the results for the ROS code

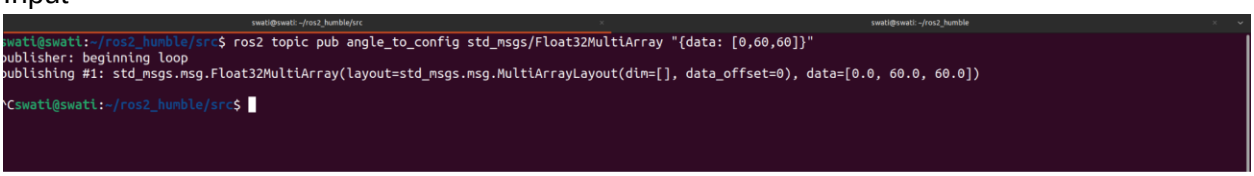
ROS code

```
import rclpy
from rclpy.node import Node
import math
from std_msgs.msg import Float32MultiArray



def angle_to_config(q_angles):
        ## initialize link variables
        l1=1
        l2=1
        l3=1
        ## initialize angle variables with converion of q angles to radians from degree
        q1=math.radians(q_angles[0])
        q2=math.radians(q_angles[1])
        q3=math.radians(q_angles[2])

        ## calculate robot's position vector using forward kinematics
        x = (l3 * math.cos(q1)* math.cos(q2) * math.cos(q3)) - (l3 * math.cos(q1) * math.sin(q2) * math.sin(q3))
        y = (l3 * math.sin(q1)* math.cos(q3)) - (l2* math.sin(q1)* math.sin(q2) * math.sin(q3))
        z = -(l3 * math.sin(q2)* math.cos(q3)) - (l3* math.cos(q2) * math.sin(q3)) - (l2* math.sin(q2)) + l1

        ## calculate rotation matrix components using forward kinematics
        r11 = (math.cos(q1)* math.cos(q2) * math.cos(q3) )- (math.cos(q1)* math.sin(q2) * math.sin(q3))
        r12 = -(math.cos(q1)* math.cos(q2) * math.sin(q3) ) - (math.cos(q1)* math.sin(q2) * math.cos(q3))
        r13 = -math.sin(q1)

        r21 = (math.cos(q3) * math.sin(q1)) - (math.sin(q1) * math.sin(q2) * math.sin(q3))
        r22 = -(math.sin(q1) * math.sin(q3)) - (math.sin(q1) * math.sin(q2) * math.cos(q3))
        r23 = math.cos(q1)

        r31 = -(math.sin(q2) * math.cos(q3)) - (math.cos(q2) * math.sin(q3))
        r32 = (math.sin(q2) * math.sin(q3)) - (math.cos(q2) * math.cos(q3))
        r33 = 0


        return [r11,r12,r13, r21,r22,r23, r31,r32,r33, x, y,z]


class my_node(Node):

    def __init__(self):
        super().__init__('my_node')
        self.subscription = self.create_subscription(
            Float32MultiArray,
            'angle_to_config',
            self.listener_callback,
            10)
        self.subscription  # prevent unused variable warning

    def listener_callback(self, msg):
        configurations = angle_to_config(msg.data)
        self.get_logger().info('Robot position vector is: "%s"' % configurations[9:12])
        self.get_logger().info('Robot rotation matrix is: "%s"'% configurations[0:9])
```

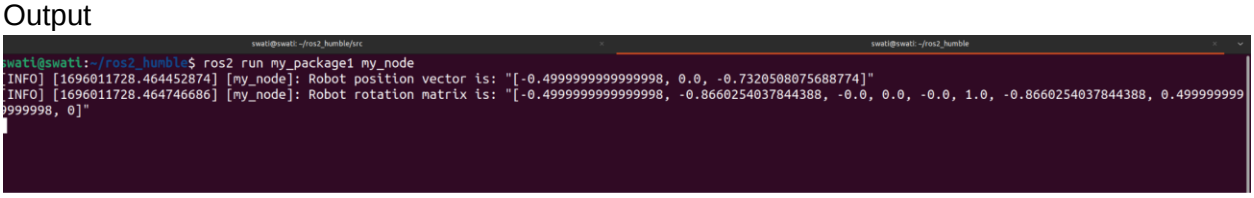
```python
class my_node(Node):

    def __init__(self):
        super().__init__('my_node')
        self.subscription = self.create_subscription(
            Float32MultiArray,
            'angle_to_config',
            self.listener_callback,
            10)
        self.subscription  # prevent unused variable warning

    def listener_callback(self, msg):
        configurations = angle_to_config(msg.data)
        self.get_logger().info('Robot position vector is: "%s"' % configurations[9:12])
        self.get_logger().info('Robot rotation matrix is: "%s"'% configurations[0:9])


def main(args=None):
    rclpy.init(args=args)
    my_node_subscriber = my_node()
    rclpy.spin(my_node_subscriber)

    # Destroy the node explicitly
    # (optional - otherwise it will be done automatically
    # when the garbage collector destroys the node object)
    my_node_subscriber.destroy_node()
    rclpy.shutdown()


if __name__ == '__main__':
    main()
```

Case 1

Input



Output



Case 2:

```
swati@swati: ~/ros2_humble/src                swati@swati: ~/ros2_humble
swati@swati:~/ros2_humble/src$ ros2 topic pub angle_to_config std_msgs/Float32MultiArray "{data: [0,100,80]}"
publisher: beginning loop
publishing #1: std_msgs.msg.Float32MultiArray(layout=std_msgs.msg.MultiArrayLayout(dim=[], data_offset=0), data=[0.0, 100.0, 80.0])

^Cswati@swati:~/ros2_humble/src$
```

Output

```
swati@swati: ~/ros2_humble/src                swati@swati: ~/ros2_humble
swati@swati:~/ros2_humble$ ros2 run my_package1 my_node
[INFO] [1696011803.793605735] [my_node]: Robot position vector is: "[-0.9999999999999999, 0.0, 0.015192246987791869]"
[INFO] [1696011803.793893366] [my_node]: Robot rotation matrix is: "[-0.9999999999999999, -8.326672684688674e-17, -0.0, 0.0, -0.0, 1.0, -8.326672684688674e-17, 0.999
9999999999999, 0]"
```

Case 3

```
swati@swati: ~/ros2_humble/src                swati@swati: ~/ros2_humble
swati@swati:~/ros2_humble/src$ ros2 topic pub angle_to_config std_msgs/Float32MultiArray "{data: [90,120,45]}"
publisher: beginning loop
publishing #1: std_msgs.msg.Float32MultiArray(layout=std_msgs.msg.MultiArrayLayout(dim=[], data_offset=0), data=[90.0, 120.0, 45.0])

^Cswati@swati:~/ros2_humble/src$
```

Output

```
swati@swati: ~/ros2_humble/src                swati@swati: ~/ros2_humble
swati@swati:~/ros2_humble$ ros2 run my_package1 my_node
[INFO] [1696011876.524011786] [my_node]: Robot position vector is: "[-5.914589856893348e-17, 0.0947343454907531, -0.12484444888695978]"
[INFO] [1696011876.524320748] [my_node]: Robot rotation matrix is: "[-5.914589856893348e-17, -1.5848095757158837e-17, -1.0, 0.0947343454907531, -1.3194792168823422,
6.123233995736766e-17, -0.258819045102521, 0.9659258262890681, 0]"
```