I. (10 Points) Write an NQC program that allows the rotation sensor assembly shown below to emulate a touch sensor. Assume that the rotation sensor is connected to input Port#1. The RCX should play a sound (of your choice) if the lever is pulled.

```nqc
task main ()
{
    SetSensorType (SENSOR_1, SENSOR_TYPE_ROTATION);
    ClearSensor (SENSOR_1);

    while (true)
    {
        if(SENSOR_1 > 1 || SENSOR_1 < -1)
        {
            PlaySound (SOUND_DOUBLE_BEEP);
            Wait(50);
        }
    }
}
```
II. (12 points) Write an NQC program that allows the following touch sensor multiplexer to steer TankBot. Assume the multiplexer has a long connector and is connected to Port #1. Your program should allow you to driver TankBot as follows: if switch#1=1 & switch#2=1, drive forward; if switch#1=0 & switch#2=0, stop; if switch#1=1 & switch#2=0, turn right; if switch#1=0 & switch#2=1, turn left.

<table>
<thead>
<tr>
<th>Switch#1</th>
<th>Switch#2</th>
<th>Safe NQC Ranges</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>RAW \geq 1000</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>43 &lt; RAW &lt; 60</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>70 &lt; RAW &lt; 200</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>RAW &lt; 42</td>
</tr>
</tbody>
</table>

```c

task main ()
{
  SetSensorType (SENSOR_1, SENSOR_TYPE_TOUCH);
  SetSensorMode (SENSOR_1, SENSOR_MODE_RAW);

  while (true)
  {
    if (SENSOR_1 > 1000) Off (OUT_A+OUT_C);
    if (SENSOR_1 < 42) OnFwd (OUT_A+OUT_C);
    if (SENSOR_1 > 70 && SENSOR_1 < 200) (OnFwd (OUT_A); OnRev (OUT_C));
    if (SENSOR_1 > 43 && SENSOR_1 < 60) (OnRev (OUT_A); OnFwd (OUT_C));
    
  }
}
```
III. (12 points) Write an NQC program that allows the following touch sensor system to emulate a rotation sensor. Your program should support positive and negative (i.e., clockwise and counter clockwise rotation) readings. Assume the touch sensor is connected to Port #1 and that a motor connected to OUT_A drives the cam’s axle.
Solution with uni-directional readings (insensitive to direction of rotation):

```cpp
task main ()
{
    SetSensorType (SENSOR_1, SENSOR_TYPE_TOUCH);
    SetSensorMode (SENSOR_1, SENSOR_MODE_PULSE);

    while (true)
    {
        OnFwd(OUT_A);
        SetUserDisplay (SENSOR_1,0);
    }
}
```

Solution with bi-directional reading (sensitive to the direction of rotation):

```cpp
int x;

// Code follows as in the previous solution
```