HINTS TO QUESTION 2 b):

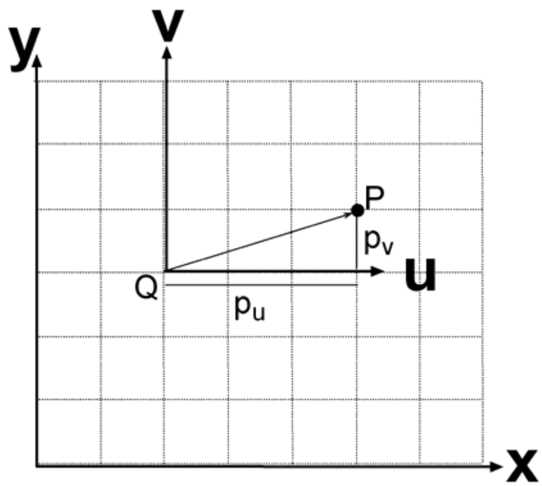
**Why do we need to change coordinate systems to perform mirroring?**

**Answer:** Draw yourself a 2D coordinate system and then a mirror not aligned with the axes and not centered at the origin. Now take an arbitrary point **p** and try to compute the mirror image of it. Then try to determine a single matrix performing this transformation. You will recognize very soon that the computation is very complex – and it's even more complex in 3D.
However, we have seen in the lecture that it is very easy to reflect a point on a coordinate axes (a coordinate plane in 3D) – this is achieved by a simple scaling where one of the scaling factors is -1. Hence if we can transform a point into mirror coordinates we can compute the mirrored point by a simple reflection of one of its coordinates.
So the whole transformation looks like this: $\mathbf{M}_{\text{mirror}} = \mathbf{R}_{\text{uvn\_to\_xyz}}\,\mathbf{S}_{\text{reflect\_n\_coordinate}}\,\mathbf{R}_{\text{xyx\_to\_uvn}}$

Have a look at the code and you will see that all you have to so is implementing two of these matrices :P

Now let's have another look how to transform a point from one coordinate system to another one:

**Translation only:**




Given is a uv-coordinate system with origin $\mathbf{q} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ (where **q** is the position vector of the point Q) and the basis vectors $\mathbf{u} = \mathbf{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\mathbf{v} = \mathbf{y} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. The point P in xy-coordinates is $\mathbf{p} = \begin{pmatrix} 5 \\ 4 \end{pmatrix}$. We want to find the uv-coordinates of this point.

As indicated in the figure above you can find these coordinates by simply subtracting the origin of the uv-coordinate system, i.e. $\mathbf{p}_{\text{uv-coordinates}} = \mathbf{p}_{\text{xy-coordinates}} - \mathbf{q} = \begin{pmatrix} 1 & 0 & -q_x \\ 0 & 1 & -q_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}$.

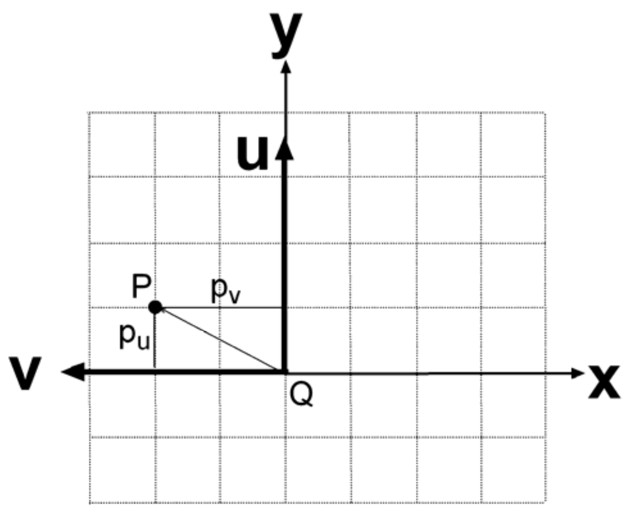
So we can see that it's a simple translation by –**q**.

In the above example (using 2D coordinates) we get

$$\begin{pmatrix} p_u \\ p_v \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \end{pmatrix} - \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

which is indeed the correct answer :-)

**Rotation only:**




Given is a uv-coordinate system with origin $\mathbf{q} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ (where **q** is the position vector of the point Q) and the basis vectors $\mathbf{u} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $\mathbf{v} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$, i.e. the uv-coordinate system is obtained by rotating the xy-coordinate system by 90 degree.

The point P in xy-coordinates is $\mathbf{p} = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$. We want to find the uv-coordinates of this point.

As indicated in the figure above you can find these coordinates by projecting the vector **p** onto the u-axis and v-axis.

$p_u = \mathbf{p} \bullet \mathbf{u} = p_x u_x + p_y u_y$ and $p_v = \mathbf{p} \bullet \mathbf{v} = p_x v_x + p_y v_y$

or in matrix form $\begin{pmatrix} p_u \\ p_v \\ 1 \end{pmatrix} = \begin{pmatrix} u_x & u_y & 0 \\ v_x & v_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}$

We can see that this is just the rotation matrix which rotates the uv-coordinate system into the xy-coordinate system (i.e. it's the **inverse** of the matrix in chapter 5, slide 40 of the handouts)!

In the above example (using 2D coordinates) we get

$$\begin{pmatrix} p_u \\ p_v \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} -2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

which is indeed the correct answer :-)

**The general case:**

In the general case you have to perform a rotation and a translation as shown above. You only have to figure out which one comes first :-)
In question 1 you deal with 3x3 matrices, hence you will have to use 4x4 matrices if you use homogenous coordinates. However, note that you don't need homogenous coordinates: say you want to transform the point **p** first by a translation and then a rotation, then you can compute this as **R** (**p**+**t**) where **R** is a 3x3 rotation matrix and **t** is a translation vector. Hence you can use the methods in the `Geometry` library in order to solve this question.

**NOTES:**

1. We can see from the above that in order to transform a point from xy-coordinates into uv-coordinates we have to transform the uv-coordinate system into the xy-coordinate system, i.e transforming a point from coordinate system 1 into coordinate system 2 is equivalent with transforming the coordinate system 2 into the coordinate system 1. This is the reason why we have a negative translation vector and the inverse of the rotation matrix we talked about in the lecture.

2. If we want to transform a point from uv-coordinates to xy-coordinates then the opposite would be the case.

3. The same principles apply in question 1, except that this time you have to define a 4x4 matrix doing the coordinate transformation which is then used by me to perform the mirroring in OpenGL.