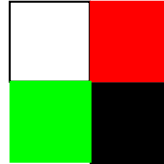**Computer Science**   COMPSCI 372 S2 C – Exercise Sheet 7
Sample Solution

Q1: Give is a 2x2 pixel big texture image with the pixel colours

White (255, 255, 255)	Red (255, 0, 0)
Green (0, 255, 0)	black (0,0,0)

The image is mapped onto a polygon which occupies 3x3 pixels on the screen.

(a) What is the colour of the screen pixels if **nearest neighbour interpolation** is used? Assume that if two distances are the same the first pixel in image order (left to right, bottom to top) is used.

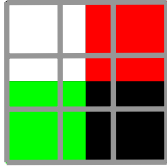**Solution:** The situation we are facing can be illustrated as follows:
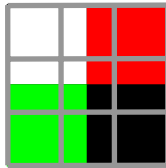
The pixels at the corners lie entirely within a texel of the texture image and hence get the corresponding colour. For the pixels which are partially covered by two or more texels of the texture image we compute the distances of the pixel centre to the centres of the corresponding texels and choose the texel closest to it.
For example, the centre of the pixel in the top-middle is closest to the centre of the white and red texel. In this case we choose the white texel since in the image array it comes before the red pixel. The resulting image on the screen is:
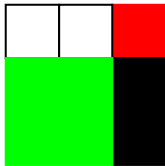
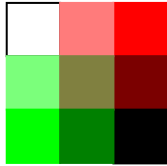(b) What is the colour of the screen pixels if **linear interpolation** is used?

Solution: The situation we are facing can be illustrated as follows:

The pixels at the corners lie entirely within a texel of the texture image and hence get the corresponding colour. For the other pixels we compute the distances to the texels covering part of it and use the weighted average of the texel colours.

For example, the pixel in the middle of the top row is covered to equal parts by the white and the red texel and hence the resulting colour is the average of them, i.e. 0.5*(255, 255, 255)+0.5*(255,0,0)=(255, 123, 123)=pink

The resulting image on the screen is:

Q2: Give is a 3x3 pixel big texture image with the pixel colours

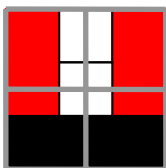| Red (255, 0, 0)	White (255, 255, 255)	Red (255, 0, 0)
Red (255, 0, 0)	White (255, 255, 255)	Red (255, 0, 0)
Black (0, 0, 0)	Black (0, 0, 0)	Black (0, 0, 0)
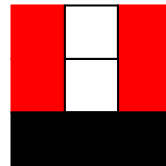
The image is mapped onto a polygon which occupies 2x2 pixels on the screen.

(a) What is the colour of the screen pixels if **nearest neighbour interpolation** is used?

Solution: The situation we are facing can be illustrated as follows:

When using nearest neighbour interpolation the colour of each pixel is the texel whose centre is closest to the centre of the pixel. The result is:

(b) What is the colour of the screen pixels if **linear interpolation** is used?

Solution: When using linear interpolation the pixel colour is the weighted average of the texels covering that pixel. The weighting factors depend on the texel distances, but we can say they are roughly proportional to the proportion of each texel covering that pixel. For example, the left top pixel is 2/3 covered by red texels and 1/3 covered by white texels and hence its colour is:

$1/3*(255, 255, 255)+2/3*(255,0,0)=(255, 85, 85)=$ reddish pink

Similarly the two bottom pixels will get a reddish black colour since most of their area is covered by black texels.

The resulting image on the screen is:

Q3: Given is the texture shown in the image below on the left and three polygons shown in the image below in the middle. Assume the texture coordinate of the bottom left vertex is (0,0) and the texture parameters GL_REPEAT are set. Which texture coordinates do we have to define for the vertex A in order to get the image shown below on the right?
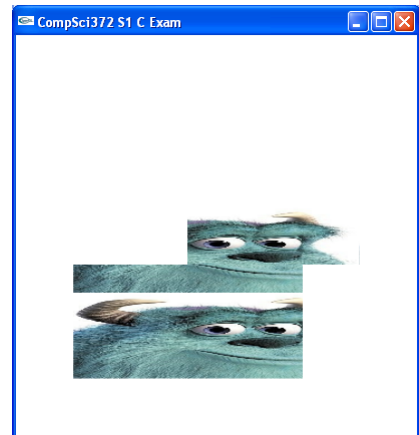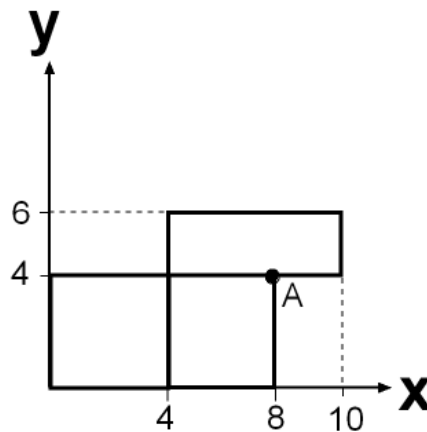

Solution: (0.8, 1.3333)

Q4: Given is the texture shown in the image below on the left and four polygons shown in the image below in the middle. Assume the following texture parameters are set:

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, **GL_REPEAT**);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, **GL_REPEAT**);

If the vertex A has the texture coordinates (0,0) which texture coordinates do you have to define for the vertices B and C in order to get the image shown below on the right?
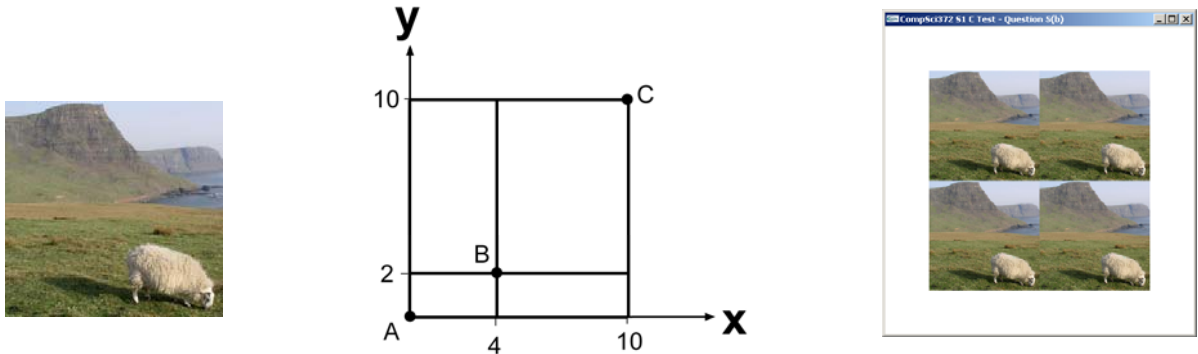

Solution: Texture coordinate for C is (2, 2). Texture coordinate for A is (0.8, 0.4)

Q5: Given is the parametric surface $p(s,t) = (4t + 2s, \sin 2\pi t, 2s)$ where $0 \le s, t \le 1$ and a function

```
float* point(float s, float t)
```

which implements $p(s,t)$ and returns an array of three floats.

Complete the code below which draws the surface below and repeats the texture image (shown on the right) <u>four times in the t parameter direction and two times in the s parameter direction</u>.
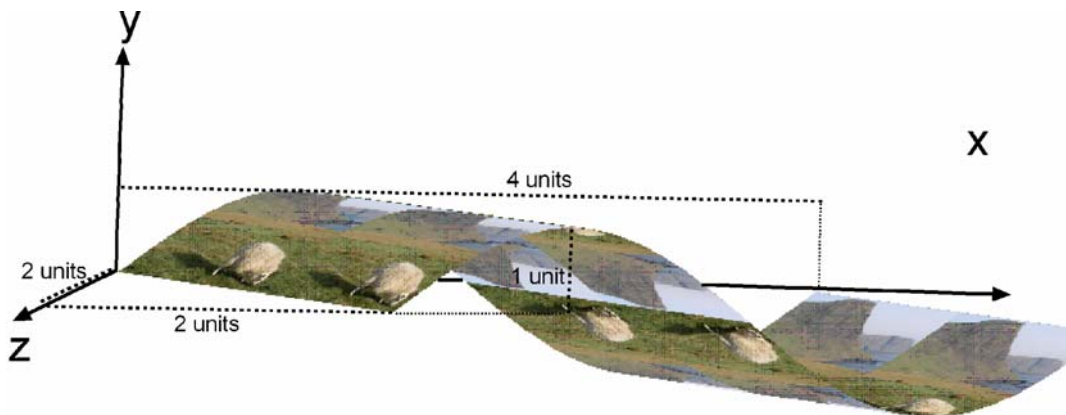
```
const int NUM_STEPS_T=32;
const int NUM_STEPS_S=16;

void display(void)
{
        // setting up the camera
        glMatrixMode( GL_MODELVIEW ); // Set the view matrix ...
        glLoadIdentity();             // ... to identity.
        gluLookAt(0,0,50, 2.5,0,0, 0,1,0); // camera is on the z-axis
        trackball.tbMatrix();         // rotate the cylinder using
                                      // the trackball ...

        glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

        // selecting the texture and enabling texture mapping.
        // NOTE: Assume that the texture wrap parameters are set to
        // GL_REPEAT.

        glBindTexture(GL_TEXTURE_2D, texName);
        glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);
        glEnable(GL_TEXTURE_2D);
```

```
        // draw the surface and with suitable texture coordinates

        float s,t1,t2;
        for (int j=0; j< NUM_STEPS_T; j++)
        {
                t1=(float) j/(float) NUM_STEPS_T;
                t2=(float) (j+1)/(float) NUM_STEPS_T;
                glBegin(GL_QUAD_STRIP);
                for (int i=0; i<= NUM_STEPS_S; i++)
                {
                        s=(float) i/(float) NUM_STEPS_S;
                        glTexCoord2f(2*s,4*t1);
                        glVertex3fv(point(s,t1));
                        glTexCoord2f(2*s,4*t2);
                        glVertex3fv(point(s,t2));
                }
                glEnd();
        }
```

```
        glDisable(GL_TEXTURE_2D);
        glFlush ();
        glutSwapBuffers();
}
```