```c
/*  checker.c
 *  This program texture maps a checkerboard image onto
 *  several rectangles, demonstrating mappings between texture
 *  coordinates and object coordinates.
 *
 *  If running this program on OpenGL 1.0, texture objects are
 *  not used.
 */


#include <GL/glut.h>
#include <stdlib.h>
#include <stdio.h>

/*      Create checkerboard texture     */
#define checkImageWidth 64
#define checkImageHeight 64
static GLubyte checkImage[checkImageHeight][checkImageWidth][4];

#ifdef GL_VERSION_1_1
static GLuint texName;
#endif

void makeCheckImage(void)
{
   int i, j, c;

   for (i = 0; i < checkImageHeight; i++) {
      for (j = 0; j < checkImageWidth; j++) {
         c = (((i&0x8)==0)^((j&0x8)==0))*255;
         checkImage[i][j][0] = (GLubyte) c;
         checkImage[i][j][1] = (GLubyte) c;
         checkImage[i][j][2] = (GLubyte) c;
         checkImage[i][j][3] = (GLubyte) 255;
      }
   }
}

void init(void)
{
   glClearColor (0.0, 0.0, 0.0, 0.0);
   glShadeModel(GL_FLAT);
   glEnable(GL_DEPTH_TEST);

   makeCheckImage();
   glPixelStorei(GL_UNPACK_ALIGNMENT, 1);

#ifdef GL_VERSION_1_1
   printf("OpenGL 1.1\n");
   glGenTextures(1, &texName);
   glBindTexture(GL_TEXTURE_2D, texName);
#else
   printf("OpenGL 1.0\n");
#endif

   glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
   glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
   glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,
                   GL_NEAREST);
   glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,
                   GL_NEAREST);
#ifdef GL_VERSION_1_1
   glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, checkImageWidth,
                checkImageHeight, 0, GL_RGBA, GL_UNSIGNED_BYTE,
                checkImage);
#else
   glTexImage2D(GL_TEXTURE_2D, 0, 4, checkImageWidth,
                checkImageHeight, 0, GL_RGBA, GL_UNSIGNED_BYTE,
                checkImage);
#endif
}

void display(void)
{
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
   glEnable(GL_TEXTURE_2D);
   glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL);
#ifdef GL_VERSION_1_1
   glBindTexture(GL_TEXTURE_2D, texName);
#endif

   glBegin(GL_QUADS);
   // Lower left
   glTexCoord2f(0.0, 0.0); glVertex3f(-2.0, -1.0, 0.0);
   glTexCoord2f(0.0, 1.0); glVertex3f(-2.0, 1.0, 0.0);
   glTexCoord2f(1.0, 1.0); glVertex3f(0.0, 1.0, 0.0);
   glTexCoord2f(1.0, 0.0); glVertex3f(0.0, -1.0, 0.0);

   // Upper left.
   glTexCoord2f(0.0, 0.0); glVertex3f(-2.0, 1.5f, 0.0);
   glTexCoord2f(0.0, 1.0); glVertex3f(-2.0, 2.5f, 0.0);
   glTexCoord2f(1.0, 1.0); glVertex3f(-1.0, 2.5f, 0.0);
   glTexCoord2f(1.0, 0.0); glVertex3f(-1.0, 1.5f, 0.0);

   // Upper middle.
   glTexCoord2f(0.0, 0.0); glVertex3f(0.0, 1.5f, 0.0);
   glTexCoord2f(0.0, 2.0); glVertex3f(0.0, 2.5f, 0.0);
   glTexCoord2f(2.0, 2.0); glVertex3f(1.0, 2.5f, 0.0);
   glTexCoord2f(2.0, 0.0); glVertex3f(1.0, 1.5f, 0.0);

   // Upper right
   glTexCoord2f(0.0, 0.0); glVertex3f(1.5f, 1.5f, 0.0);
   glTexCoord2f(0.0, 0.5); glVertex3f(1.5f, 2.5f, 0.0);
   glTexCoord2f(0.5, 0.5); glVertex3f(2.5f, 2.5f, 0.0);
   glTexCoord2f(0.5, 0.0); glVertex3f(2.5f, 1.5f, 0.0);

   // Lower right.
   glTexCoord2f(0.0, 0.0); glVertex3f(1.0, -1.0, 0.0);
   glTexCoord2f(0.0, 1.0); glVertex3f(1.0, 1.0, 0.0);
   glTexCoord2f(1.0, 1.0); glVertex3f(2.41421f, 1.0, -1.41421f);
   glTexCoord2f(1.0, 0.0); glVertex3f(2.41421f, -1.0, -1.41421f);
   glEnd();
   glFlush();
   glDisable(GL_TEXTURE_2D);
}

void reshape(int w, int h)
{
   glViewport(0, 0, (GLsizei) w, (GLsizei) h);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluPerspective(60.0, (GLfloat) w/(GLfloat) h, 1.0, 30.0);
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
   //glTranslatef(0.0, 0.0, -3.6f);
   glTranslatef(0.0, 0.0, -5.0f);
}

void keyboard (unsigned char key, int x, int y)
{
   switch (key) {
      case 27:
         exit(0);
         break;
      default:
         break;
   }
}

int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
   glutInitWindowSize(400, 400);
   glutInitWindowPosition(100, 100);
   glutCreateWindow(argv[0]);
   init();
   glutDisplayFunc(display);
   glutReshapeFunc(reshape);
   glutKeyboardFunc(keyboard);
   glutMainLoop();
   return 0;
}
```