```
191 void drawCylinder() {
192     {
193         double radiiTop = halfRadius;
194         double radiiBase = halfRadius*.80;
195         double height = cylinderHeight;
196         int sliceNum = 30;
197         int stackNum = 30;
198         glPushMatrix();
199         {
200             glTranslated(0.0, -( height/2.0 ), 0);
201             glRotated(-90, 1.0, 0, 0);
202             GLUquadricObj *myCylinder = gluNewQuadric();
203             gluQuadricDrawStyle(myCylinder, GLU_FILL);
204             gluCylinder(myCylinder, radiiBase, radiiTop, height, sliceNum, stackNum);
205         }
206         glPopMatrix();
207     }
208 }
209
210 void drawBranches(double tiltAngle, double xOffset, double yOffset) {
211     glPushMatrix();
212     {
213         //Draw right branch
214         glTranslated(xOffset, yOffset, 0.0);
215         glRotated(tiltAngle, 0, 0, 1.0);
216         drawCylinder();
217     }
218     glPopMatrix();
219     glPushMatrix();
220     {
221         //draw left branch
222         glTranslated(-xOffset, yOffset, 0.0);
223
224         glRotated(-tiltAngle, 0, 0, 1.0);
225
226         drawCylinder();
227     }
228     glPopMatrix();
```

```
229 }
230
231 void drawBranchesRecursive(int countLeft, double tiltAngle, double xOffset, double yOffset) {
232     //countleft==the number of levels of branches above this one.
233     if( countLeft == 0) {
234         return;
235     }
236     glPushMatrix();
237     {
238         //Draw right branch
239         glTranslated(xOffset, yOffset, 0.0);
240
241         glRotated(-tiltAngle, 0, 0, 1.0);
242         drawCylinder();
243
244         //glPushMatrix();
245         //{
246             drawBranchesRecursive(countLeft-1, tiltAngle * .75, xOffset, yOffset);
247         //}
248         //glPopMatrix();
249     }
250     glPopMatrix();
251     glPushMatrix();
252     {
253         //draw left branch
254         glTranslated(-xOffset, yOffset, 0.0);
255
256         glRotated(tiltAngle, 0, 0, 1.0);
257
258         drawCylinder();
259         //glPushMatrix();
260         //{
261             drawBranchesRecursive(countLeft-1, tiltAngle * .75, xOffset, yOffset);
262         //}
263         //glPopMatrix();
264     }
265     glPopMatrix();
266 }
```

```
267
268 void Timer(int value) {
269
270    growTreeByVal( deltaTreeGrow );
271
272    glutPostRedisplay();      // Post re-paint request to activate display()
273    glutTimerFunc(refreshMills, Timer, deltaTreeGrow ); // next Timer call milliseconds later
274 }
275
276 void pauseMine() {
277    if ( deltaTreeGrow == 0 ) {
278       deltaTreeGrow = 1;
279    } else {
280       deltaTreeGrow = 0;
281    }
282 }
283
284 void display() {
285
286    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
287
288    glMatrixMode(GL_MODELVIEW);
289    //save current matrix
290    glPushMatrix();
291
292     double eyex = 0.0;   //1.0
293     double eyey = 0.0;   //1.0
294     double eyez = 10.0;   //1.5
295    //set up the camera
296    gluLookAt(eyex, eyey, eyez, //eye position
297              0.0, 0.0, 0.0, //point we are looking at
298              0.0, 1.0, 0.0);  //up vector
299
300    //position the light in the scene
301    pos_light();
302    //draw the cube
303    //drawcube();
304
```

```
305     drawCylinder();
306 
307      double halfRadius = .25;
308      double cylinderHeight = 2.0;
309 
310      drawBranchesRecursive(treeHeight, 45, halfRadius, cylinderHeight*.75);
311 
312      /*
313      for (int i = 1; i < 50; i++) {
314          //for (int i = 0; i < 50; i++) {
315          //drawBranches(-30.0, i*halfRadius+i*halfRadius, (i+1)*(cylinderHeight) );
316          drawBranches(-30.0, i*halfRadius, i*(cylinderHeight) );
317 
318      }
319       */
320 
321     glPopMatrix();
322 
323     glutSwapBuffers();
324 
325 }
326 
327 void reshape(int w, int h) {
328     GW = w;
329     GH = h;
330 
331     glMatrixMode(GL_PROJECTION);
332     glLoadIdentity();
333     //notice the change in the near and far planes - they are measure with respect to the
334     //camera position
335     glOrtho( -20*(float)w/h, (float)20*w/h, -5, 20, 1.0, 15.0);
336     glMatrixMode(GL_MODELVIEW);
337     glViewport(0, 0, w, h);
338 
339 }
340 
341 
342 void keyboard(unsigned char key, int x, int y )
```