**Lab Taks-4**

Submission Guidelines-

* Rename the file to your id only. If your id is 18-XXXXX-1, then the file name must be 18-XXXXX-1.docx.
* Must submit within time that will be discussed in class VUES to the section named Lab Tak-4
* Must include resources for all the section in the table

|  |
| --- |
| **Question- 1**  Draw the scenario of a traffic signal using function to represent each object |
| Graph Plot (Picture)-  //not showing the points value otherwise the graph can not be understand for too many points |
| **Code-**  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  /\* Handler for window-repaint event. Call back when the window first appears and  whenever the window needs to be re-painted. \*/  void back\_jungles(){  //extreme back jungles  glBegin(GL\_POLYGON);  glColor3f(0.1, 0.6, 0);  glVertex2f(-40, -16);  glVertex2f(-40, 2.2);  glVertex2f(-38, 4);  glVertex2f(-32.5, 5.98);  glVertex2f(-17.6, 11.2);  glVertex2f(4, 15);  glVertex2f(29, 11.5);  glVertex2f(45, -1);  glVertex2f(45, -14);  glEnd();  // back small jungles  glBegin(GL\_POLYGON);  glColor3f(0, 0.9, 0);  glVertex2f(-40, -10);  glVertex2f(-38, -8);  glVertex2f(-36, -6.706921);  glVertex2f(-32.9358216, -6.706921);  glVertex2f(-31, -8);  glVertex2f(-29, -10);  glVertex2f(-27, -6);  glVertex2f(-25, -3);  glVertex2f(-21, -2);  glVertex2f(-19, -3);  glVertex2f(-17, -6);  glVertex2f(-14, -10);  glVertex2f(-12, -11);  glVertex2f(-11, -9);  glVertex2f(-8, -9);  glVertex2f(-7, -10);  glVertex2f(-6, -8);  glVertex2f(-5, -5);  glVertex2f(-2,-4);  glVertex2f(0,-4);  glVertex2f(3,-7);  glVertex2f(4.5,-7.5);  glVertex2f(5,-8.5);  glVertex2f(10,-5);  glVertex2f(14,-4.5);  glVertex2f(16,-5.5);  glVertex2f(17.5,-7.5);  glVertex2f(19,-10.5);  glVertex2f(21.5,-8.5);  glVertex2f(25,-7);  glVertex2f(28.5,-6.5);  glVertex2f(32.5,-8);  glVertex2f(34.5,-11.5);  glVertex2f(36.5,-13);  glVertex2f(38.5,-10);  glVertex2f(41,-8.5);  glVertex2f(45,-8);  glVertex2f(45, -16);  glVertex2f(-40, -16);  glEnd();  }  void traffic\_poll(){  //black area under traffic light poll  glBegin(GL\_POLYGON);  glColor3f(0, 0, 0);  glVertex2f(-37, -14);  glVertex2f(-37, -16);  glVertex2f(45, -16);  glVertex2f(45, -14);  glEnd();  // light poll -- black area  glBegin(GL\_POLYGON);  glColor3f(0, 0, 0);  glVertex2f(-36, 2);  glVertex2f(-36, -14);  glVertex2f(-34.5, -14);  glVertex2f(-34.5, 2);  glEnd();  // light gray area of poll  glBegin(GL\_POLYGON);  glColor3f(0.3, 0.3, 0.3);  glVertex2f(-38, 12);  glVertex2f(-38, 2);  glVertex2f(-32.5, 2);  glVertex2f(-32.5, 12);  glEnd();  }  void three\_circle(float radius, float xc, float yc, float r, float g, float b)  {  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3f(r,g,b);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=radius;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+xc,y+yc);  }  glEnd();  }  void road(){  //road  glBegin(GL\_POLYGON);  glColor3f(0.3, 0.3, 0.3);  glVertex2f(-40, -16);  glVertex2f(-40, -30);  glVertex2f(45, -30);  glVertex2f(45, -16);  glEnd();  //white zebra crossing signs  //road  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex2f(-36, -16.7);  glVertex2f(-36.5, -18.2);  glVertex2f(-25.5, -18.2);  glVertex2f(-25, -16.7);  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(-36.5, -19);  glVertex2f(-37, -20.5);  glVertex2f(-26, -20.5);  glVertex2f(-25.5, -19);  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(-37, -22);  glVertex2f(-37.5, -23.5);  glVertex2f(-26.5, -23.5);  glVertex2f(-26, -22);  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(-37.5, -25);  glVertex2f(-38, -26.5);  glVertex2f(-27, -26.5);  glVertex2f(-26.5, -25);  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(-38, -28);  glVertex2f(-38.5, -29.5);  glVertex2f(-27.5, -29.5);  glVertex2f(-27, -28);  glEnd();  //two under the car  glBegin(GL\_POLYGON);  glVertex2f(-2, -22);  glVertex2f(-3, -23);  glVertex2f(5, -23);  glVertex2f(6, -22);  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(16, -22);  glVertex2f(15, -23);  glVertex2f(24, -23);  glVertex2f(24.8, -22);  glEnd();  }  void car(){  //red body  glBegin(GL\_POLYGON);  glColor3f(1, 0, 0);  glVertex2f(-8.2, -19.8);  glVertex2f(-8.2, -19.8);  glVertex2f(-13, -19.2);  glVertex2f(-13.2, -17.4);  glVertex2f(-13.1, -16.8);  glVertex2f(-12.8, -15.8);  glVertex2f(-4, -14);  glVertex2f(-2, -10);  glVertex2f(10, -10);  glVertex2f(16.75, -15.95);  glVertex2f(16, -20);  glVertex2f(14.414494, -20.0239526);  glEnd();  //black sqaure  glBegin(GL\_POLYGON);  glColor3f(0, 0, 0);  glVertex2f(15, -16);  glVertex2f(16, -16);  glVertex2f(16, -17);  glVertex2f(15, -17);  glEnd();  //window 1  glBegin(GL\_POLYGON);  glColor3f(0, 0, 0);  glVertex2f(0, -11);  glVertex2f(-2, -15);  glVertex2f(4, -15);  glVertex2f(4, -11);  glEnd();  //window 2 square  glBegin(GL\_POLYGON);  glColor3f(0, 0, 0);  glVertex2f(5, -11);  glVertex2f(5, -15);  glVertex2f(10, -15);  glVertex2f(10, -11);  glEnd();  //wheel 1 in between  glBegin(GL\_POLYGON);  glLineWidth(2.5);  for(int i=0;i<560;i++)  {  glColor3f(0.3,0.3,0.3);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r= 2.88844;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x - 5.3779922367174,y - 19.2035973808484);  }  glEnd();  //wheel\_1  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<360;i++)  {  glColor3f(0,0,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r= 2.3809527;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x - 5.3779922367174,y - 19.2035973808484 );  }  //glVertex2f(0.3f,0.4f);  //glVertex2f(0.1f,0.4f);  glEnd();  //wheel 1 outline  glBegin(GL\_POINTS);  glLineWidth(2.5);  for(int i=0;i<560;i++)  {  glColor3f(0,0,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r= 2.88844;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x - 5.3779922367174,y - 19.2035973808484);  }  glEnd();  //wheel 2 in between  glBegin(GL\_POLYGON);  glLineWidth(2.5);  for(int i=0;i<560;i++)  {  glColor3f(0.3,0.3,0.3);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r= 2.730296;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x + 11.7782074159104,y - 19.3136665772639);  }  glEnd();  //wheel 2  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<360;i++)  {  glColor3f(0,0,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r= 2.184420;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x + 11.7782074159104,y - 19.3136665772639);  }  //glVertex2f(0.3f,0.4f);  //glVertex2f(0.1f,0.4f);  glEnd();  //wheel 2 outline  glBegin(GL\_POINTS);  glLineWidth(2.5);  for(int i=0;i<560;i++)  {  glColor3f(0,0,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r= 2.730296;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x + 11.7782074159104,y - 19.3136665772639);  }  glEnd();  }  void sky()  {  //cloud 2  glBegin(GL\_POLYGON);  glColor3f(1, 1, 1);  glVertex2f(33.8, 19.6);  glVertex2f(33.8, 16.8);  glVertex2f(38.2, 16.8);  glVertex2f(40.6, 18);  glVertex2f(40.6, 20);  glVertex2f(39.4, 21.6);  glVertex2f(36.4, 21.8);  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(20, 20);  glVertex2f(20, 16.8);  glVertex2f(33.8, 16.8);  glVertex2f(33.8, 19.6);  glVertex2f(31.2, 21.3);  glVertex2f(28.2, 22.8);  glVertex2f(23.4, 22.8);  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(20, 20);  glVertex2f(18.4, 21.4);  glVertex2f(15, 22.8);  glVertex2f(12.8, 21.6);  glVertex2f(12.8, 18.8);  glVertex2f(15.4, 16.8);  glVertex2f(20.6, 16.8);  glEnd();  //cloud 1  glBegin(GL\_POLYGON);  glVertex2f(-23, 20.5);  glVertex2f(-25, 22);  glVertex2f(-30.5, 23);  glVertex2f(-35, 21);  glVertex2f(-34, 18);  glVertex2f(-30, 17);  glVertex2f(-23, 16.5);  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(-23, 20.5);  glVertex2f(-23, 16.5);  glVertex2f(-18, 17);  glVertex2f(-14, 19);  glVertex2f(-14.5, 21.5);  glVertex2f(-16.5, 23);  glVertex2f(-20.5, 22);  glVertex2f(-20.5, 22);  glEnd();  //birds  glLineWidth(4);  glColor3f(0,0,0);  glBegin(GL\_LINES);  glVertex2f(-6, 22.6);  glVertex2f(-5.2, 23);  glVertex2f(-5.2, 23);  glVertex2f(-4.4, 22.6);  glVertex2f(-4.4, 22.6);  glVertex2f(-3.6, 21.8);  glVertex2f(-3.6, 21.8);  glVertex2f(-2.9, 20.8);  glVertex2f(-2.9, 20.8);  glVertex2f(-2.4, 21.8);  glVertex2f(-2.4, 21.8);  glVertex2f(-1.6, 22.6);  glVertex2f(-1.6, 22.6);  glVertex2f(-0.8, 23);  glVertex2f(-0.8, 23);  glVertex2f(0 ,22.6);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-2, 20.6);  glVertex2f(-1.2, 21);  glVertex2f(-1.2, 21);  glVertex2f(-0.4, 20.6);  glVertex2f(-0.4, 20.6);  glVertex2f(0.4, 19.8);  glVertex2f(0.4, 19.8);  glVertex2f(1.1, 18.8);  glVertex2f(1.1, 18.8);  glVertex2f(1.6, 19.8);  glVertex2f(1.6, 19.8);  glVertex2f(2.4, 20.6);  glVertex2f(2.4, 20.6);  glVertex2f(3.2, 21);  glVertex2f(3.2, 21);  glVertex2f(4, 20.6);  glEnd();  glBegin(GL\_LINES);  glVertex2f(1, 23.6);  glVertex2f(1.8, 24);  glVertex2f(1.8, 24);  glVertex2f(2.6, 23.6);  glVertex2f(2.6, 23.6);  glVertex2f(3.4, 22.8);  glVertex2f(3.4, 22.8);  glVertex2f(4.1, 21.8);  glVertex2f(4.1, 21.8);  glVertex2f(4.6, 22.8);  glVertex2f(4.6, 22.8);  glVertex2f(5.4, 23.6);  glVertex2f(5.4, 23.6);  glVertex2f(6.2, 24);  glVertex2f(6.2, 24);  glVertex2f(7, 23.6);  glEnd();  }  void tree(){  glColor3f(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(24, -4);  glVertex2f(24, -14.037031);  glVertex2f(26, -14.037031);  glVertex2f(26, -4);  glEnd();  glColor3f(0,0.4,0);  glBegin(GL\_POLYGON);  glVertex2f(25, 3);  glVertex2f(16, -4);  glVertex2f(34, -4);  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(18, 0);  glVertex2f(32, 0);  glVertex2f(25,5.5);  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(20, 3);  glVertex2f(30, 3);  glVertex2f(25,8);  glEnd();  }  void display() {  glClearColor(0.0f, 0.6f,0.90f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)  back\_jungles();  traffic\_poll();  three\_circle( 1.18558802,-35.2276891476581, 10.3308657487617,1, 0, 0);  three\_circle( 1.18558802,-35.2276891476581, 7.3308657,1, 1, 0);  three\_circle( 1.18558802,-35.2276891476581, 4.0308657,0, 1, 0);  road();  car();  sky();  tree();  glFlush(); // Render now  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInitWindowSize(1320, 820); // Set the window's initial width & height  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title  glutDisplayFunc(display); // Register display callback handler for window re-paint  gluOrtho2D(-40, +45, -30, 25);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 2**  Draw two village scenarios for day and night using function to represent each object |
| **Graph Plot (Picture)-**  **Day**    **Night** |
| **Code-**  ***//Day scenario code***  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  /\* Handler for window-repaint event. Call back when the window first appears and  whenever the window needs to be re-painted. \*/  void khor\_pala(){  //upper extension of the khor pala  glColor3f(0, 0, 0);  glBegin(GL\_POLYGON);  //right yellow chad  glVertex2f(-433.98356, 57.3442926);  glVertex2f(-434, 80);  glVertex2f(-425, 80);  glVertex2f(-425, 57.3442926);  glEnd();  //yellow khor pala  glColor3f(1, 1, 0);  glBegin(GL\_POLYGON);  glVertex2f(-387.48784, 23.76182488);  glVertex2f(-400, 45);  glVertex2f(-430, 60);  glVertex2f(-468, 34);  glVertex2f(-484, 6);  glVertex2f(-490, -52);  glVertex2f(-486, -78);  glVertex2f(-455, -100);  glVertex2f(-405, -100);  glVertex2f(-380, -80);  glVertex2f(-380, -20);  glEnd();  }  void house\_2(){  //right yellow chad  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  glVertex2f(-365, 50);  glVertex2f(-320, -20);  glVertex2f(-165, -20);  glVertex2f(-225, 50);  glEnd();  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  //left yellow chad  glVertex2f(-365, 50);  glVertex2f(-425, -20);  glVertex2f(-380, -20);  glVertex2f(-347.9209, 23.43264033);  glEnd();  //under 1 gray color area (door)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-347.9209, 23.43264033);  glVertex2f(-380, -20);  glVertex2f(-380, -80);  glVertex2f(-320, -90);  glVertex2f(-320, -20);  glEnd();  //under 2 gray color area (window)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-320, -20);  glVertex2f(-320, -90);  glVertex2f(-185, -65);  glVertex2f(-185, -20);  glEnd();  //under extension of house 2  // number 1(door)  glColor3f(0.9, 0.8, 0.7);  glBegin(GL\_POLYGON);  glVertex2f(-380, -80);  glVertex2f(-405, -100);  glVertex2f(-320, -115);  glVertex2f(-320, -90);  glEnd();  // number 2(extension)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-320, -90);  glVertex2f(-320, -115);  glVertex2f(-170, -80);  glVertex2f(-185, -65);  glEnd();  //door  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  glVertex2f(-360, -30);  glVertex2f(-360, -70);  glVertex2f(-340, -70);  glVertex2f(-340, -30);  glEnd();  //window  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  glVertex2f(-280, -30);  glVertex2f(-280, -60);  glVertex2f(-250, -60);  glVertex2f(-250, -30);  glEnd();  }  void house\_1(){  //black area under traffic light poll  //upper portion  glColor3f(1, 0.7, 0);  glBegin(GL\_POLYGON);  glVertex2f(-455, 50);  glVertex2f(-555, 50);  glVertex2f(-650, -25);  glVertex2f(-490, -25);  glVertex2f(-440, 0);  glEnd();  //lowerportion  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-620.0788546, -25);  glVertex2f(-619.975899, -78);  glVertex2f(-486, -78);  glVertex2f(-486, -25);  glEnd();  //extream lower portion  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex2f(-636, -78);  glVertex2f(-636, -100);  glVertex2f(-455, -100);  glVertex2f(-455, -78);  glEnd();  //door  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  glVertex2f(-570, -30);  glVertex2f(-570, -70);  glVertex2f(-540, -70);  glVertex2f(-540, -30);  glEnd();  }  void grass(){  glColor3f(0, 1, 0);  glBegin(GL\_POLYGON);  glVertex2f(-1000, -75);  glVertex2f(-1000, -100);  glVertex2f(-710, -140);  glVertex2f(-765, -180);  glVertex2f(-425, -230);  glVertex2f(-445, -250);  glVertex2f(-435, -265);  glVertex2f(-380, -275);  glVertex2f(-315, -280);  glVertex2f(-150, -300);  glVertex2f(50, -300);  glVertex2f(50, -40);  glEnd();  }  void tree(){  glBegin(GL\_POLYGON);  glColor3f(0.0f, 1.0f, 0.0f);  glVertex2f(-140, 30.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-168, 16.5);  glVertex2f(-179, 27);  glVertex2f(-180, 40);  glVertex2f(-189, 44);  glVertex2f(-197.2, 53.4);  glVertex2f(-200.2, 68);  glVertex2f(-197.5,80.5);  glVertex2f(-192.5,87.5);  glVertex2f(-184.3,91.3);  glVertex2f(-182.6,101.1);  glVertex2f(-176,109);  glVertex2f(-166,111.5);  glVertex2f(-155.8,111.85 );  glVertex2f(-150.8,110.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-140.4,114 );  glVertex2f(-129.8,117.6 );  glVertex2f(-119.05,117.35 );  glVertex2f(-111.1,113.7 );  glVertex2f(-104.4,106.2 );  glVertex2f(-100,100 );  glVertex2f(-92,100.6 );  glVertex2f(-82,97.4 );  glVertex2f(-76.1,88.8);  glVertex2f(-76,78.8);  glVertex2f(-78.45, 71.9);  glVertex2f(-73.6, 67.8);  glVertex2f(-70.2, 61.4);  glVertex2f(-71, 51);  glVertex2f(-75.86, 43.68);  glVertex2f(-73.04, 37.78);  glVertex2f(-70, 30);  glVertex2f(-73, 19);  glVertex2f(-85, 8.5);  glVertex2f(-99, 7.2);  glVertex2f(-112, 10);  glVertex2f(-120, 24.5);  glEnd();  glColor3f(0.5f, 0.3f, 0.0f);  glBegin(GL\_POLYGON);  /\*  glVertex2f(-120, 24.5);  glVertex2f(-140, 24.5);  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, -24.5);  \*/  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, 24.5);  glVertex2f(-140, 24.5);  /\*  \*/  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(-120, 24.5);  glVertex2f(-107.7, 35);  glVertex2f(-113.7, 42);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 43);  glVertex2f(-140.2, 50.6);  glVertex2f(-144.7, 42);  glVertex2f(-140, 37);  glVertex2f(-140, 24.5);  glEnd();  }  void outline\_tree(){  glColor3f(0, 0, 0);  glLineWidth(1.5);  glBegin(GL\_LINES);  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, -65);  glVertex2f(-120, 24.5);  glVertex2f(-120, 24.5);  glVertex2f(-107.7, 35);  glVertex2f(-107.7, 35);  glVertex2f(-113.7, 42);  glVertex2f(-113.7, 42);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 61);  glVertex2f(-119.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 43);  glVertex2f(-131.7, 43);  glVertex2f(-140.2, 50.6);  glVertex2f(-140.2, 50.6);  glVertex2f(-144.7, 42);  glVertex2f(-144.7, 42);  glVertex2f(-140, 37);  glVertex2f(-140, 37);  glVertex2f(-140, 24.5);  glVertex2f(-140, 24.5);  glVertex2f(-140, -65);  //round green leafs  glVertex2f(-140, 30.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-168, 16.5);  glVertex2f(-168, 16.5);  glVertex2f(-179, 27);  glVertex2f(-179, 27);  glVertex2f(-180, 40);  glVertex2f(-180, 40);  glVertex2f(-189, 44);  glVertex2f(-189, 44);  glVertex2f(-197.2, 53.4);  glVertex2f(-197.2, 53.4);  glVertex2f(-200.2, 68);  glVertex2f(-200.2, 68);  glVertex2f(-197.5,80.5);  glVertex2f(-197.5,80.5);  glVertex2f(-192.5,87.5);  glVertex2f(-192.5,87.5);  glVertex2f(-184.3,91.3);  glVertex2f(-184.3,91.3);  glVertex2f(-182.6,101.1);  glVertex2f(-182.6,101.1);  glVertex2f(-176,109);  glVertex2f(-176,109);  glVertex2f(-166,111.5);  glVertex2f(-166,111.5);  glVertex2f(-155.8,111.85 );  glVertex2f(-155.8,111.85 );  glVertex2f(-150.8,110.2 );  glVertex2f(-150.8,110.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-140.4,114 );  glVertex2f(-140.4,114 );  glVertex2f(-129.8,117.6 );  glVertex2f(-129.8,117.6 );  glVertex2f(-119.05,117.35 );  glVertex2f(-119.05,117.35 );  glVertex2f(-111.1,113.7 );  glVertex2f(-111.1,113.7 );  glVertex2f(-104.4,106.2 );  glVertex2f(-104.4,106.2 );  glVertex2f(-100,100 );  glVertex2f(-100,100 );  glVertex2f(-92,100.6 );  glVertex2f(-92,100.6 );  glVertex2f(-82,97.4 );  glVertex2f(-82,97.4 );  glVertex2f(-76.1,88.8);  glVertex2f(-76.1,88.8);  glVertex2f(-76,78.8);  glVertex2f(-76,78.8);  glVertex2f(-78.45, 71.9);  glVertex2f(-78.45, 71.9);  glVertex2f(-73.6, 67.8);  glVertex2f(-73.6, 67.8);  glVertex2f(-70.2, 61.4);  glVertex2f(-70.2, 61.4);  glVertex2f(-71, 51);  glVertex2f(-71, 51);  glVertex2f(-75.86, 43.68);  glVertex2f(-75.86, 43.68);  glVertex2f(-73.04, 37.78);  glVertex2f(-73.04, 37.78);  glVertex2f(-70, 30);  glVertex2f(-70, 30);  glVertex2f(-73, 19);  glVertex2f(-73, 19);  glVertex2f(-85, 8.5);  glVertex2f(-85, 8.5);  glVertex2f(-99, 7.2);  glVertex2f(-99, 7.2);  glVertex2f(-112, 10);  glVertex2f(-112, 10);  glVertex2f(-120, 16.5);  //glVertex2f(-120, 24.5);  glEnd();  }  void two\_hills(){  // first hill  glColor3f(0, 0.8, 0);  glBegin(GL\_POLYGON);  glVertex2f(-1000, -45.5);  glVertex2f(-900, 50);  glVertex2f(-883, 61);  glVertex2f(-870, 61);  glVertex2f(-850, 51);  glVertex2f(-812.251844, 26.727090);  glVertex2f(-895, -75);  glVertex2f(-1000, -75);  glEnd();  //second hill  glBegin(GL\_POLYGON);  glColor3f(0, 0.5, 0);  glVertex2f(-895, -75);  //glVertex2f(-812.251844, 26.727090);  glVertex2f(-786, 59);  glVertex2f(-780, 65);  glVertex2f(-773, 65);  glVertex2f(-764, 61);  glVertex2f(-580, -75);  glEnd();  }  void sun(){  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<360;i++)  {  glColor3f(1,0.9,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r= 85.3901037;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x - 824.535315,y + 53.8050655 );  }  //glVertex2f(0.3f,0.4f);  //glVertex2f(0.1f,0.4f);  glEnd();  glLineWidth(5);  glColor3f(0, 0, 0);  glBegin(GL\_LINES);  glVertex2f(-968, 48);  glVertex2f(-926, 56);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-980, 88);  glVertex2f(-918, 70);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-954, 124);  glVertex2f(-922, 104);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-954, 124);  glVertex2f(-922, 104);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-925, 150);  glVertex2f(-896, 124);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-888,158);  glVertex2f(-874,138);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-852, 194);  glVertex2f(-850, 150);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-814, 170);  glVertex2f(-816, 152);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-758, 172);  glVertex2f(-780, 142);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-744, 142);  glVertex2f(-760, 128);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-680, 150);  glVertex2f(-740, 110);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-680, 100);  glVertex2f(-730, 90);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-728, 66);  glVertex2f(-674, 58);  glEnd();  }  void boat(){  //sitting area  glBegin(GL\_POLYGON);  glColor3f(0.5, 0.3, 0);  glVertex2f(-936, -218);  glVertex2f(-824, -218);  glVertex2f(-800.6849, -179.6123);  glVertex2f(-972, -180);  glEnd();  //shade  glBegin(GL\_POLYGON);  glColor3f(1, .9, .6);  glVertex2f(-880, -194);  glVertex2f(-836, -194);  glVertex2f(-817.9758973, -186.61688);  glVertex2f(-826.261432, -169.25207);  glVertex2f(-850, -148);  glVertex2f(-907, -148);  glVertex2f(-894, -157);  glVertex2f(-885.1914705, -180.2796);  glEnd();  //shade left  glBegin(GL\_POLYGON);  glColor3f(1, .9, .6);  glVertex2f(-936.8605, -180.1132086);  glVertex2f(-885.1914705, -180.2796);  glVertex2f(-894, -157);  glVertex2f(-907, -148);  glVertex2f(-924, -156);  glVertex2f(-934, -174);  glEnd();  //stick  glLineWidth(6);  glBegin(GL\_LINES);  glColor3f(0, 0, 0);  glVertex2f(-846, -240);  glVertex2f(-806, -106);  glEnd();  //black lower area  glBegin(GL\_POLYGON);  glColor3f(0, 0, 0);  glVertex2f(-936, -218);  glVertex2f(-824, -218);  glVertex2f(-782, -172);  glVertex2f(-836, -194);  glVertex2f(-930, -194);  glVertex2f(-972, -180);  glEnd();  }  void display() {  glClearColor(0.0f, 0.6f,0.90f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT);  sun();  two\_hills();  grass(); // Clear the color buffer (background)  tree();  house\_1();  khor\_pala();  house\_2();  boat();  glFlush(); // Render now  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInitWindowSize(1020, 520); // Set the window's initial width & height  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title  glutDisplayFunc(display); // Register display callback handler for window re-paint  gluOrtho2D(-1000, +50, -300, 200);  glutMainLoop(); // Enter the event-processing loop  return 0;  }  ***//Night scenario code***  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  /\* Handler for window-repaint event. Call back when the window first appears and  whenever the window needs to be re-painted. \*/  void khor\_pala(){  //upper extension of the khor pala  glColor3f(0, 0, 0);  glBegin(GL\_POLYGON);  //right yellow chad  glVertex2f(-433.98356, 57.3442926);  glVertex2f(-434, 80);  glVertex2f(-425, 80);  glVertex2f(-425, 57.3442926);  glEnd();  //yellow khor pala  glColor3f(.9, 0.5, 0);  glBegin(GL\_POLYGON);  glVertex2f(-387.48784, 23.76182488);  glVertex2f(-400, 45);  glVertex2f(-430, 60);  glVertex2f(-468, 34);  glVertex2f(-484, 6);  glVertex2f(-490, -52);  glVertex2f(-486, -78);  glVertex2f(-455, -100);  glVertex2f(-405, -100);  glVertex2f(-380, -80);  glVertex2f(-380, -20);  glEnd();  }  void house\_2(){  //right yellow chad  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  glVertex2f(-365, 50);  glVertex2f(-320, -20);  glVertex2f(-165, -20);  glVertex2f(-225, 50);  glEnd();  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  //left yellow chad  glVertex2f(-365, 50);  glVertex2f(-425, -20);  glVertex2f(-380, -20);  glVertex2f(-347.9209, 23.43264033);  glEnd();  //under 1 gray color area (door)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-347.9209, 23.43264033);  glVertex2f(-380, -20);  glVertex2f(-380, -80);  glVertex2f(-320, -90);  glVertex2f(-320, -20);  glEnd();  //under 2 gray color area (window)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-320, -20);  glVertex2f(-320, -90);  glVertex2f(-185, -65);  glVertex2f(-185, -20);  glEnd();  //under extension of house 2  // number 1(door)  glColor3f(0.9, 0.8, 0.7);  glBegin(GL\_POLYGON);  glVertex2f(-380, -80);  glVertex2f(-405, -100);  glVertex2f(-320, -115);  glVertex2f(-320, -90);  glEnd();  // number 2(extension)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-320, -90);  glVertex2f(-320, -115);  glVertex2f(-170, -80);  glVertex2f(-185, -65);  glEnd();  //door  glColor3f(1, 1, 0);  glBegin(GL\_POLYGON);  glVertex2f(-360, -30);  glVertex2f(-360, -70);  glVertex2f(-340, -70);  glVertex2f(-340, -30);  glEnd();  //window  glColor3f(1, 1, 0);  glBegin(GL\_POLYGON);  glVertex2f(-280, -30);  glVertex2f(-280, -60);  glVertex2f(-250, -60);  glVertex2f(-250, -30);  glEnd();  }  void house\_1(){  //black area under traffic light poll  //upper portion  glColor3f(1, 0.7, 0);  glBegin(GL\_POLYGON);  glVertex2f(-455, 50);  glVertex2f(-555, 50);  glVertex2f(-650, -25);  glVertex2f(-490, -25);  glVertex2f(-440, 0);  glEnd();  //lowerportion  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-620.0788546, -25);  glVertex2f(-619.975899, -78);  glVertex2f(-486, -78);  glVertex2f(-486, -25);  glEnd();  //extream lower portion  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex2f(-636, -78);  glVertex2f(-636, -100);  glVertex2f(-455, -100);  glVertex2f(-455, -78);  glEnd();  //door  glColor3f(1, 1, 0);  glBegin(GL\_POLYGON);  glVertex2f(-570, -30);  glVertex2f(-570, -70);  glVertex2f(-540, -70);  glVertex2f(-540, -30);  glEnd();  }  void grass(){  glColor3f(0, 1, 0);  glBegin(GL\_POLYGON);  glVertex2f(-1000, -75);  glVertex2f(-1000, -100);  glVertex2f(-710, -140);  glVertex2f(-765, -180);  glVertex2f(-425, -230);  glVertex2f(-445, -250);  glVertex2f(-435, -265);  glVertex2f(-380, -275);  glVertex2f(-315, -280);  glVertex2f(-150, -300);  glVertex2f(50, -300);  glVertex2f(50, -40);  glEnd();  }  void tree(){  glBegin(GL\_POLYGON);  glColor3f(0.0f, 1.0f, 0.0f);  glVertex2f(-140, 30.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-168, 16.5);  glVertex2f(-179, 27);  glVertex2f(-180, 40);  glVertex2f(-189, 44);  glVertex2f(-197.2, 53.4);  glVertex2f(-200.2, 68);  glVertex2f(-197.5,80.5);  glVertex2f(-192.5,87.5);  glVertex2f(-184.3,91.3);  glVertex2f(-182.6,101.1);  glVertex2f(-176,109);  glVertex2f(-166,111.5);  glVertex2f(-155.8,111.85 );  glVertex2f(-150.8,110.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-140.4,114 );  glVertex2f(-129.8,117.6 );  glVertex2f(-119.05,117.35 );  glVertex2f(-111.1,113.7 );  glVertex2f(-104.4,106.2 );  glVertex2f(-100,100 );  glVertex2f(-92,100.6 );  glVertex2f(-82,97.4 );  glVertex2f(-76.1,88.8);  glVertex2f(-76,78.8);  glVertex2f(-78.45, 71.9);  glVertex2f(-73.6, 67.8);  glVertex2f(-70.2, 61.4);  glVertex2f(-71, 51);  glVertex2f(-75.86, 43.68);  glVertex2f(-73.04, 37.78);  glVertex2f(-70, 30);  glVertex2f(-73, 19);  glVertex2f(-85, 8.5);  glVertex2f(-99, 7.2);  glVertex2f(-112, 10);  glVertex2f(-120, 24.5);  glEnd();  glColor3f(0.5f, 0.3f, 0.0f);  glBegin(GL\_POLYGON);  /\*  glVertex2f(-120, 24.5);  glVertex2f(-140, 24.5);  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, -24.5);  \*/  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, 24.5);  glVertex2f(-140, 24.5);  /\*  \*/  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(-120, 24.5);  glVertex2f(-107.7, 35);  glVertex2f(-113.7, 42);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 43);  glVertex2f(-140.2, 50.6);  glVertex2f(-144.7, 42);  glVertex2f(-140, 37);  glVertex2f(-140, 24.5);  glEnd();  }  void outline\_tree(){  glColor3f(0, 0, 0);  glLineWidth(1.5);  glBegin(GL\_LINES);  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, -65);  glVertex2f(-120, 24.5);  glVertex2f(-120, 24.5);  glVertex2f(-107.7, 35);  glVertex2f(-107.7, 35);  glVertex2f(-113.7, 42);  glVertex2f(-113.7, 42);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 61);  glVertex2f(-119.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 43);  glVertex2f(-131.7, 43);  glVertex2f(-140.2, 50.6);  glVertex2f(-140.2, 50.6);  glVertex2f(-144.7, 42);  glVertex2f(-144.7, 42);  glVertex2f(-140, 37);  glVertex2f(-140, 37);  glVertex2f(-140, 24.5);  glVertex2f(-140, 24.5);  glVertex2f(-140, -65);  //round green leafs  glVertex2f(-140, 30.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-168, 16.5);  glVertex2f(-168, 16.5);  glVertex2f(-179, 27);  glVertex2f(-179, 27);  glVertex2f(-180, 40);  glVertex2f(-180, 40);  glVertex2f(-189, 44);  glVertex2f(-189, 44);  glVertex2f(-197.2, 53.4);  glVertex2f(-197.2, 53.4);  glVertex2f(-200.2, 68);  glVertex2f(-200.2, 68);  glVertex2f(-197.5,80.5);  glVertex2f(-197.5,80.5);  glVertex2f(-192.5,87.5);  glVertex2f(-192.5,87.5);  glVertex2f(-184.3,91.3);  glVertex2f(-184.3,91.3);  glVertex2f(-182.6,101.1);  glVertex2f(-182.6,101.1);  glVertex2f(-176,109);  glVertex2f(-176,109);  glVertex2f(-166,111.5);  glVertex2f(-166,111.5);  glVertex2f(-155.8,111.85 );  glVertex2f(-155.8,111.85 );  glVertex2f(-150.8,110.2 );  glVertex2f(-150.8,110.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-140.4,114 );  glVertex2f(-140.4,114 );  glVertex2f(-129.8,117.6 );  glVertex2f(-129.8,117.6 );  glVertex2f(-119.05,117.35 );  glVertex2f(-119.05,117.35 );  glVertex2f(-111.1,113.7 );  glVertex2f(-111.1,113.7 );  glVertex2f(-104.4,106.2 );  glVertex2f(-104.4,106.2 );  glVertex2f(-100,100 );  glVertex2f(-100,100 );  glVertex2f(-92,100.6 );  glVertex2f(-92,100.6 );  glVertex2f(-82,97.4 );  glVertex2f(-82,97.4 );  glVertex2f(-76.1,88.8);  glVertex2f(-76.1,88.8);  glVertex2f(-76,78.8);  glVertex2f(-76,78.8);  glVertex2f(-78.45, 71.9);  glVertex2f(-78.45, 71.9);  glVertex2f(-73.6, 67.8);  glVertex2f(-73.6, 67.8);  glVertex2f(-70.2, 61.4);  glVertex2f(-70.2, 61.4);  glVertex2f(-71, 51);  glVertex2f(-71, 51);  glVertex2f(-75.86, 43.68);  glVertex2f(-75.86, 43.68);  glVertex2f(-73.04, 37.78);  glVertex2f(-73.04, 37.78);  glVertex2f(-70, 30);  glVertex2f(-70, 30);  glVertex2f(-73, 19);  glVertex2f(-73, 19);  glVertex2f(-85, 8.5);  glVertex2f(-85, 8.5);  glVertex2f(-99, 7.2);  glVertex2f(-99, 7.2);  glVertex2f(-112, 10);  glVertex2f(-112, 10);  glVertex2f(-120, 16.5);  //glVertex2f(-120, 24.5);  glEnd();  }  void two\_hills(){  // first hill  glColor3f(0, 0.8, 0);  glBegin(GL\_POLYGON);  glVertex2f(-1000, -45.5);  glVertex2f(-900, 50);  glVertex2f(-883, 61);  glVertex2f(-870, 61);  glVertex2f(-850, 51);  glVertex2f(-812.251844, 26.727090);  glVertex2f(-895, -75);  glVertex2f(-1000, -75);  glEnd();  //second hill  glBegin(GL\_POLYGON);  glColor3f(0, 0.5, 0);  glVertex2f(-895, -75);  //glVertex2f(-812.251844, 26.727090);  glVertex2f(-786, 59);  glVertex2f(-780, 65);  glVertex2f(-773, 65);  glVertex2f(-764, 61);  glVertex2f(-580, -75);  glEnd();  }  void moon(){  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3f(1,1,1);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r= 41.4766199;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x - 500,y + 140);  }  //glVertex2f(0.3f,0.4f);  //glVertex2f(0.1f,0.4f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1,1,1);  glVertex2f(-430, 174);  glVertex2f(-436, 166);  glVertex2f(-436, 186);  glVertex2f(-420, 172);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1,1,1);  glVertex2f(-430, 174);  glVertex2f(-444, 176);  glVertex2f(-422, 186);  glEnd();  //star 2  glBegin(GL\_POLYGON);  glColor3f(1,1,1);  glVertex2f(-430, 152);  glVertex2f(-438, 136);  glVertex2f(-430, 142);  glVertex2f(-422, 136);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1,1,1);  glVertex2f(-430, 142);  glVertex2f(-440, 146);  glVertex2f(-420, 146);  glEnd();  //star 3  glBegin(GL\_POLYGON);  glColor3f(1,1,1);  glVertex2f(-399.5, 164.5);  glVertex2f(-403.5, 156.5);  glVertex2f(-400, 159.5);  glVertex2f(-396.5, 156.5);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(1,1,1);  glVertex2f(-400, 159.5);  glVertex2f(-404, 162.5);  glVertex2f(-395, 162.5);  glEnd();  }  void night\_sky(){  glBegin(GL\_POLYGON);  glColor3f(0, 0.1, 0);  glVertex2f(-1000, 200);  glVertex2f(-1000,-75);  glVertex2f(50, -40.00177);  glVertex2f(50, 200);  glEnd();  }  void boat(){  //sitting area  glBegin(GL\_POLYGON);  glColor3f(0.5, 0.3, 0);  glVertex2f(-936, -218);  glVertex2f(-824, -218);  glVertex2f(-800.6849, -179.6123);  glVertex2f(-972, -180);  glEnd();  //shade  glBegin(GL\_POLYGON);  glColor3f(1, .9, .6);  glVertex2f(-880, -194);  glVertex2f(-836, -194);  glVertex2f(-817.9758973, -186.61688);  glVertex2f(-826.261432, -169.25207);  glVertex2f(-850, -148);  glVertex2f(-907, -148);  glVertex2f(-894, -157);  glVertex2f(-885.1914705, -180.2796);  glEnd();  //shade left  glBegin(GL\_POLYGON);  glColor3f(1, .9, .6);  glVertex2f(-936.8605, -180.1132086);  glVertex2f(-885.1914705, -180.2796);  glVertex2f(-894, -157);  glVertex2f(-907, -148);  glVertex2f(-924, -156);  glVertex2f(-934, -174);  glEnd();  //stick  glLineWidth(6);  glBegin(GL\_LINES);  glColor3f(0, 0, 0);  glVertex2f(-846, -240);  glVertex2f(-806, -106);  glEnd();  //black lower area  glBegin(GL\_POLYGON);  glColor3f(0, 0, 0);  glVertex2f(-936, -218);  glVertex2f(-824, -218);  glVertex2f(-782, -172);  glVertex2f(-836, -194);  glVertex2f(-930, -194);  glVertex2f(-972, -180);  glEnd();  }  void display() {  glClearColor(0.0f, 0.6f,0.90f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT);  night\_sky();  moon();  two\_hills();  grass(); // Clear the color buffer (background)  tree();  house\_1();  khor\_pala();  house\_2();  boat();  glFlush(); // Render now  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInitWindowSize(1020, 520); // Set the window's initial width & height  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title  glutDisplayFunc(display); // Register display callback handler for window re-paint  gluOrtho2D(-1000, +50, -300, 200);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-**  **Day**    **Night** |