**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)-** |
| Code-  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  #include <vector>  void sky()  {  glColor3ub(135, 206, 235);  glBegin(GL\_POLYGON);  glVertex2f(-10,0);glVertex2f(5,0);  glVertex2f(5,5);glVertex2f(-10,5);  glEnd();  }  void \_signals(float radius, float xC, float yC, float r, float g, float b) {  glBegin(GL\_POLYGON);  for(int i=0; i<200; i++){  glColor3ub(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 signalbox()  { glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(-5.6,1.5);glVertex2f(-6.4,1.5);  glVertex2f(-6.4,3.8);glVertex2f(-5.6,3.8);  glEnd();  }  void signalhandle()  { glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(-6.2,0);glVertex2f(-6.2,1.5);  glVertex2f(-5.8,1.5);glVertex2f(-5.8,0);  glEnd();  }  void road() {  glColor3ub(192, 189, 179);  glBegin(GL\_POLYGON);  glVertex2f(-10, 0);glVertex2f(5, 0);  glVertex2f(5,-2);glVertex2f(-10, -2);  glEnd();  }  void car1()  {  glColor3ub(120, 81, 169);  glBegin(GL\_POLYGON);  glVertex2f(-5.4,0.2);glVertex2f(-5.4,0.8);  glVertex2f(-4.6,1.4);glVertex2f(-3.8,1.4);  glVertex2f(-3,0.8);glVertex2f(-3,0.2);  glEnd();  //glass  glColor3ub(216, 191, 216);  glBegin(GL\_POLYGON);  glVertex2f(-4.6,1.2);glVertex2f(-5.2,0.8);  glVertex2f(-4.6,0.8);  glEnd();  glColor3ub(216, 191, 216);  glBegin(GL\_POLYGON);  glVertex2f(-4.6,1.2);glVertex2f(-4.6,0.8);  glVertex2f(-3.6,0.8);glVertex2f(-3.8,1.2);  glEnd();  glColor3ub(216, 191, 216);  glBegin(GL\_POLYGON);  glVertex2f(-3.8,1.2);glVertex2f(-3.6,0.8);  glVertex2f(-3.2,0.8);  glEnd();  //lights  glColor3ub(255, 234, 0);  glBegin(GL\_POLYGON);  glVertex2f(-5.2,0.6);glVertex2f(-5.2,0.4);  glVertex2f(-5,0.5);  glEnd();  glColor3ub(255, 234, 0);  glBegin(GL\_POLYGON);  glVertex2f(-3.2,0.4);glVertex2f(-3.2,0.6);  glVertex2f(-3.4,0.5);  glEnd();  //tire  glColor3ub(0, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(-5.2,0);glVertex2f(-5.2,0.2);  glVertex2f(-4.8,0.2);glVertex2f(-4.8,0);  glEnd();  glColor3ub(0, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(-3.6,0.2);glVertex2f(-3.2,0.2);  glVertex2f(-3.2,0);glVertex2f(-3.6,0);  glEnd();  }  void car2()  {  glColor3ub(9, 121, 105);  glBegin(GL\_POLYGON);  glVertex2f(-1.8,1.4);glVertex2f(-0.8,1.4);  glVertex2f(-0.2,0.8);glVertex2f(-0.2,0.2);  glVertex2f(-2.4,0.2);glVertex2f(-2.4,0.8);  glEnd();  //glass  glColor3ub(175, 225, 175);  glBegin(GL\_POLYGON);  glVertex2f(-1.8,1.2);glVertex2f(-2.2,0.8);  glVertex2f(-1.8,0.8);  glEnd();  glColor3ub(175, 225, 175);  glBegin(GL\_POLYGON);  glVertex2f(-1.8,1.2);glVertex2f(-0.8,1.2);  glVertex2f(-0.8,0.8);glVertex2f(-1.8,0.8);  glEnd();  glColor3ub(175, 225, 175);  glBegin(GL\_POLYGON);  glVertex2f(-0.8,1.2);glVertex2f(-0.4,0.8);  glVertex2f(-0.8,0.8);  glEnd();  //lights  glColor3ub(255, 234, 0);  glBegin(GL\_POLYGON);  glVertex2f(-2.2,0.6);glVertex2f(-2.2,0.4);  glVertex2f(-2,0.5);  glEnd();  glColor3ub(255, 234, 0);  glBegin(GL\_POLYGON);  glVertex2f(-0.4,0.6);glVertex2f(-0.4,0.4);  glVertex2f(-0.6,0.5);  glEnd();  //tire  glColor3ub(0, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(-2.2,0);glVertex2f(-2.2,0.2);  glVertex2f(-1.8,0.2);glVertex2f(-1.8,0);  glEnd();  glColor3ub(0, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(-0.4,0.2);glVertex2f(-0.8,0.2);  glVertex2f(-0.8,0);glVertex2f(-0.4,0);  glEnd();  }  void display() {  glClearColor(1, 1, 1, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  sky();  signalbox();  signalhandle();  road();  \_signals(.3,-6,3.2,255,0,0);  \_signals(.3,-6,2.6,234, 201, 79);  \_signals(.3,-6,2,0,255,0);  car1();  car2();  glFlush(); // Render now  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutInitWindowSize(500, 320);  //glutInitWindowPosition((glutGet(GLUT\_SCREEN\_WIDTH)-1100)/2,(glutGet(GLUT\_SCREEN\_HEIGHT)-600)/2);  glutInitWindowPosition(200, 50);  glutCreateWindow("Lab Task - 4");  //this line must be below of glutCreateWindow();  gluOrtho2D(-10,3, -2, 5);  glutDisplayFunc(display);  glutMainLoop();  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)-** |
| **Code-**  For day scenario- #include <windows.h>  #include <GL/glut.h>  #include <math.h>  #include <vector>  void houseshed1()  {  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(2,2);glVertex2f(3,3);  glVertex2f(3,2);  glEnd();  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(3,3);glVertex2f(5.5,3);  glVertex2f(5.5,2);glVertex2f(3,2);  glEnd();  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(3,3);glVertex2f(5.5,3);  glVertex2f(5.5,2);glVertex2f(3,2);  glEnd();  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(5.5,3);glVertex2f(6.5,2);  glVertex2f(5.5,2);  glEnd();  }  void housebody1()  {  glColor3ub(164,90,82);  glBegin(GL\_POLYGON);  glVertex2f(2.5,2);glVertex2f(2.5,0.5);  glVertex2f(6,0.5);glVertex2f(6,2);  glEnd();  //door  glColor3ub(150,0,24);  glBegin(GL\_POLYGON);  glVertex2f(4,0.5);glVertex2f(4,1.2);  glVertex2f(4.5,1.2);glVertex2f(4.5,0.5);  glEnd();  //window left  glColor3ub(150, 0, 24);  glBegin(GL\_POLYGON);  glVertex2f(2.8,1.8);glVertex2f(2.8,1.4);  glVertex2f(3.4,1.4);glVertex2f(3.4,1.8);  glEnd();  //window right  glColor3ub(150,0,24);  glBegin(GL\_POLYGON);  glVertex2f(5,1.8);glVertex2f(5,1.4);  glVertex2f(5.6,1.4);glVertex2f(5.6,1.8);  glEnd();  }  void houseshed2()  {  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(5,3.1);glVertex2f(5.8,3.8);  glVertex2f(5.8,3.1);  glEnd();  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(5.8,3.8);glVertex2f(7.6,3.8);  glVertex2f(7.6,3.1);glVertex2f(5.8,3.1);  glEnd();  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(7.6,3.8);glVertex2f(8.3,3.1);  glVertex2f(7.6,3.1);  glEnd();  }  void housebody2()  {  glColor3ub(164,90,82);  glBegin(GL\_POLYGON);  glVertex2f(5.2,3.1);glVertex2f(5.2,3);  glVertex2f(8,3.1);glVertex2f(8,1.5);  //glVertex2f(6,1.5);  glEnd();  glColor3ub(164,90,82);  glBegin(GL\_POLYGON);  glVertex2f(6,2);glVertex2f(6,1.5);  glVertex2f(8,1.5);glVertex2f(8,2);  glEnd();  glColor3ub(164,90,82);  glBegin(GL\_POLYGON);  glVertex2f(6.5,2);glVertex2f(6.1,2.4);  glVertex2f(8,2.4);glVertex2f(8,2);  glEnd();  glColor3ub(164,90,82);  glBegin(GL\_POLYGON);  glVertex2f(6.2,2.4);glVertex2f(6.2,2.8);  glVertex2f(6.8,2.8);glVertex2f(6.8,2.4);  glEnd();  //door  glColor3ub(150,0,24);  glBegin(GL\_POLYGON);  glVertex2f(6.6,1.5);glVertex2f(6.6,2.2);  glVertex2f(7,2.2);glVertex2f(7,1.5);  glEnd();  //window left  glColor3ub(150, 0, 24);  glBegin(GL\_POLYGON);  glVertex2f(5.7,2.8);glVertex2f(6.2,2.8);  glVertex2f(6.2,2.4);glVertex2f(6.1,2.4);  glEnd();  //window right  glColor3ub(150,0,24);  glBegin(GL\_POLYGON);  glVertex2f(7.2,2.8);glVertex2f(7.2,2.4);  glVertex2f(7.8,2.4);glVertex2f(7.8,2.8);  glEnd();  }  void land()  {  glColor3ub(183,224,175);  glBegin(GL\_POLYGON);  glVertex2f(0,0);glVertex2f(15,0);  glVertex2f(15,5);glVertex2f(0,5);  glEnd();  }  void tree1()  {  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(9.4,3.8);glVertex2f(8.8,3);  glVertex2f(10,3);  glEnd();  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(9.4,3.4);glVertex2f(8.8,2.6);  glVertex2f(10,2.6);  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(9.4,3.2);glVertex2f(8.8,2.2);  glVertex2f(10,2.2);  glEnd();  glColor3ub(89, 51, 32);  glBegin(GL\_POLYGON);  glVertex2f(9.5,2.2);glVertex2f(9.5,0.6);  glVertex2f(9.3,0.6);glVertex2f(9.3,2.2);  glEnd();  }  void tree2()  {  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(12,4);glVertex2f(11,3);  glVertex2f(13,3);  glEnd();  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(12,3.8);glVertex2f(11,2.6);  glVertex2f(13,2.6);  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(12,3.6);glVertex2f(11,2.2);  glVertex2f(13,2.2);  glEnd();  glColor3ub(89, 51, 32);  glBegin(GL\_POLYGON);  glVertex2f(11.8,2.2);glVertex2f(11.8,1);  glVertex2f(12.2,1);glVertex2f(12.2,2.2);  glEnd();  }  void boat()  { //below  glColor3ub(0,0,139);  glBegin(GL\_POLYGON);  glVertex2f(3,-1.5);glVertex2f(3.5,-2);  glVertex2f(3.5,-1.5);  glEnd();  glColor3ub(0,0,139);  glBegin(GL\_POLYGON);  glVertex2f(6.5,-2);glVertex2f(7,-1.5);  glVertex2f(6.5,-1.5);  glEnd();  glColor3ub(0,0,139);  glBegin(GL\_POLYGON);  glVertex2f(3.5,-2);glVertex2f(6.5,-2);  glVertex2f(6.5,-1.5);glVertex2f(3.5,-1.5);  glEnd();  //upper flag  glColor3ub(255, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(5,-1.45);glVertex2f(5,-0.8);  glVertex2f(2.9,-1.45);  glEnd();  glColor3ub(108, 122, 134);  glBegin(GL\_POLYGON);  glVertex2f(5.2,-1.5);glVertex2f(5.2,-0.8);  glVertex2f(5.24,-0.8);glVertex2f(5.24,-1.5);  glEnd();  glColor3ub(108,122,134);  glBegin(GL\_POLYGON);  glVertex2f(5.4,-1.5);glVertex2f(5.4,-0.8);  glVertex2f(5.44,-0.8);glVertex2f(5.44,-1.5);  glEnd();  }  void river()  {  glColor3ub(0,191, 255);  glBegin(GL\_POLYGON);  glVertex2f(0,-3);glVertex2f(0,0);  glVertex2f(15,0);glVertex2f(15,-3);  glEnd();  }  void hill()  {  glColor3ub(182, 91, 0);  glBegin(GL\_POLYGON);  glVertex2f(0,5);glVertex2f(2,7);  glVertex2f(4,5);  glEnd();  glColor3ub(182, 91, 0);  glBegin(GL\_POLYGON);  glVertex2f(2.5,5);glVertex2f(4,6.5);  glVertex2f(6,5);  glEnd();  glColor3ub(182, 91, 0);  glBegin(GL\_POLYGON);  glVertex2f(5,5);glVertex2f(8,8);  glVertex2f(11,5);  glEnd();  glColor3ub(182, 91, 0);  glBegin(GL\_POLYGON);  glVertex2f(10,5);glVertex2f(12,7);  glVertex2f(13,5);  glEnd();  glColor3ub(182, 91, 0);  glBegin(GL\_POLYGON);  glVertex2f(12.5,5);glVertex2f(14,6.5);  glVertex2f(15,5);  glEnd();  }  void sun()  {  glLineWidth(7.5);  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3ub(255, 222, 59);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=1.3;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+11,y+8.5);  }  //glVertex2f(0.3f,0.4f);  //glVertex2f(0.1f,0.4f);  glEnd();  }  void sky()  {  glColor3ub(116, 204, 244);  glBegin(GL\_POLYGON);  glVertex2f(0,5);glVertex2f(0,10);  glVertex2f(15,10);glVertex2f(15,5);  glEnd();  }  void display() {  glClearColor(1, 1, 1, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  land();  river();  sky();  houseshed1();  housebody1();  houseshed2();  housebody2();  tree1();  tree2();  boat();  hill();  sun();  glFlush(); // Render now  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutInitWindowSize(650, 600);  //glutInitWindowPosition((glutGet(GLUT\_SCREEN\_WIDTH)-1100)/2,(glutGet(GLUT\_SCREEN\_HEIGHT)-600)/2);  //glutInitWindowPosition();  glutCreateWindow("Lab Task - 4");  //this line must be below of glutCreateWindow();  gluOrtho2D(0, 15, -3, 10);  glutDisplayFunc(display);  glutMainLoop();  return 0;  }  For night scenario- #include <windows.h>  #include <GL/glut.h>  #include <math.h>  #include <vector>  void houseshed1()  {  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(2,2);glVertex2f(3,3);  glVertex2f(3,2);  glEnd();  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(3,3);glVertex2f(5.5,3);  glVertex2f(5.5,2);glVertex2f(3,2);  glEnd();  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(3,3);glVertex2f(5.5,3);  glVertex2f(5.5,2);glVertex2f(3,2);  glEnd();  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(5.5,3);glVertex2f(6.5,2);  glVertex2f(5.5,2);  glEnd();  }  void housebody1()  {  glColor3ub(164,90,82);  glBegin(GL\_POLYGON);  glVertex2f(2.5,2);glVertex2f(2.5,0.5);  glVertex2f(6,0.5);glVertex2f(6,2);  glEnd();  //door  glColor3ub(150,0,24);  glBegin(GL\_POLYGON);  glVertex2f(4,0.5);glVertex2f(4,1.2);  glVertex2f(4.5,1.2);glVertex2f(4.5,0.5);  glEnd();  //window left  glColor3ub(150, 0, 24);  glBegin(GL\_POLYGON);  glVertex2f(2.8,1.8);glVertex2f(2.8,1.4);  glVertex2f(3.4,1.4);glVertex2f(3.4,1.8);  glEnd();  //window right  glColor3ub(150,0,24);  glBegin(GL\_POLYGON);  glVertex2f(5,1.8);glVertex2f(5,1.4);  glVertex2f(5.6,1.4);glVertex2f(5.6,1.8);  glEnd();  }  void houseshed2()  {  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(5,3.1);glVertex2f(5.8,3.8);  glVertex2f(5.8,3.1);  glEnd();  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(5.8,3.8);glVertex2f(7.6,3.8);  glVertex2f(7.6,3.1);glVertex2f(5.8,3.1);  glEnd();  glColor3ub(217, 96, 84);  glBegin(GL\_POLYGON);  glVertex2f(7.6,3.8);glVertex2f(8.3,3.1);  glVertex2f(7.6,3.1);  glEnd();  }  void housebody2()  {  glColor3ub(164,90,82);  glBegin(GL\_POLYGON);  glVertex2f(5.2,3.1);glVertex2f(5.2,3);  glVertex2f(8,3.1);glVertex2f(8,1.5);  //glVertex2f(6,1.5);  glEnd();  glColor3ub(164,90,82);  glBegin(GL\_POLYGON);  glVertex2f(6,2);glVertex2f(6,1.5);  glVertex2f(8,1.5);glVertex2f(8,2);  glEnd();  glColor3ub(164,90,82);  glBegin(GL\_POLYGON);  glVertex2f(6.5,2);glVertex2f(6.1,2.4);  glVertex2f(8,2.4);glVertex2f(8,2);  glEnd();  glColor3ub(164,90,82);  glBegin(GL\_POLYGON);  glVertex2f(6.2,2.4);glVertex2f(6.2,2.8);  glVertex2f(6.8,2.8);glVertex2f(6.8,2.4);  glEnd();  //door  glColor3ub(150,0,24);  glBegin(GL\_POLYGON);  glVertex2f(6.6,1.5);glVertex2f(6.6,2.2);  glVertex2f(7,2.2);glVertex2f(7,1.5);  glEnd();  //window left  glColor3ub(150, 0, 24);  glBegin(GL\_POLYGON);  glVertex2f(5.7,2.8);glVertex2f(6.2,2.8);  glVertex2f(6.2,2.4);glVertex2f(6.1,2.4);  glEnd();  //window right  glColor3ub(150,0,24);  glBegin(GL\_POLYGON);  glVertex2f(7.2,2.8);glVertex2f(7.2,2.4);  glVertex2f(7.8,2.4);glVertex2f(7.8,2.8);  glEnd();  }  void land()  {  glColor3ub(134,153,61);  glBegin(GL\_POLYGON);  glVertex2f(0,0);glVertex2f(15,0);  glVertex2f(15,5);glVertex2f(0,5);  glEnd();  }  void tree1()  {  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(9.4,3.8);glVertex2f(8.8,3);  glVertex2f(10,3);  glEnd();  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(9.4,3.4);glVertex2f(8.8,2.6);  glVertex2f(10,2.6);  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(9.4,3.2);glVertex2f(8.8,2.2);  glVertex2f(10,2.2);  glEnd();  glColor3ub(89, 51, 32);  glBegin(GL\_POLYGON);  glVertex2f(9.5,2.2);glVertex2f(9.5,0.6);  glVertex2f(9.3,0.6);glVertex2f(9.3,2.2);  glEnd();  }  void tree2()  {  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(12,4);glVertex2f(11,3);  glVertex2f(13,3);  glEnd();  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(12,3.8);glVertex2f(11,2.6);  glVertex2f(13,2.6);  glColor3ub(37, 76, 29);  glBegin(GL\_POLYGON);  glVertex2f(12,3.6);glVertex2f(11,2.2);  glVertex2f(13,2.2);  glEnd();  glColor3ub(89, 51, 32);  glBegin(GL\_POLYGON);  glVertex2f(11.8,2.2);glVertex2f(11.8,1);  glVertex2f(12.2,1);glVertex2f(12.2,2.2);  glEnd();  }  void boat()  { //below  glColor3ub(0,0,139);  glBegin(GL\_POLYGON);  glVertex2f(3,-1.5);glVertex2f(3.5,-2);  glVertex2f(3.5,-1.5);  glEnd();  glColor3ub(0,0,139);  glBegin(GL\_POLYGON);  glVertex2f(6.5,-2);glVertex2f(7,-1.5);  glVertex2f(6.5,-1.5);  glEnd();  glColor3ub(0,0,139);  glBegin(GL\_POLYGON);  glVertex2f(3.5,-2);glVertex2f(6.5,-2);  glVertex2f(6.5,-1.5);glVertex2f(3.5,-1.5);  glEnd();  //upper flag  glColor3ub(255, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(5,-1.45);glVertex2f(5,-0.8);  glVertex2f(2.9,-1.45);  glEnd();  glColor3ub(108, 122, 134);  glBegin(GL\_POLYGON);  glVertex2f(5.2,-1.5);glVertex2f(5.2,-0.8);  glVertex2f(5.24,-0.8);glVertex2f(5.24,-1.5);  glEnd();  glColor3ub(108,122,134);  glBegin(GL\_POLYGON);  glVertex2f(5.4,-1.5);glVertex2f(5.4,-0.8);  glVertex2f(5.44,-0.8);glVertex2f(5.44,-1.5);  glEnd();  }  void river()  {  glColor3ub(35,137,218);  glBegin(GL\_POLYGON);  glVertex2f(0,-3);glVertex2f(0,0);  glVertex2f(15,0);glVertex2f(15,-3);  glEnd();  }  void hill()  {  glColor3ub(182, 91, 0);  glBegin(GL\_POLYGON);  glVertex2f(0,5);glVertex2f(2,7);  glVertex2f(4,5);  glEnd();  glColor3ub(182, 91, 0);  glBegin(GL\_POLYGON);  glVertex2f(2.5,5);glVertex2f(4,6.5);  glVertex2f(6,5);  glEnd();  glColor3ub(182, 91, 0);  glBegin(GL\_POLYGON);  glVertex2f(5,5);glVertex2f(8,8);  glVertex2f(11,5);  glEnd();  glColor3ub(182, 91, 0);  glBegin(GL\_POLYGON);  glVertex2f(10,5);glVertex2f(12,7);  glVertex2f(13,5);  glEnd();  glColor3ub(182, 91, 0);  glBegin(GL\_POLYGON);  glVertex2f(12.5,5);glVertex2f(14,6.5);  glVertex2f(15,5);  glEnd();  }  void moon()  {  glLineWidth(7.5);  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3f(1.0,1.0,1.0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=1.3;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+11,y+8.5);  }  //glVertex2f(0.3f,0.4f);  //glVertex2f(0.1f,0.4f);  glEnd();  }  void sky()  {  glColor3ub(46, 68, 130);  glBegin(GL\_POLYGON);  glVertex2f(0,5);glVertex2f(0,10);  glVertex2f(15,10);glVertex2f(15,5);  glEnd();  }  void display() {  glClearColor(1, 1, 1, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  land();  river();  sky();  houseshed1();  housebody1();  houseshed2();  housebody2();  tree1();  tree2();  boat();  hill();  moon();  glFlush(); // Render now  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutInitWindowSize(650, 600);  //glutInitWindowPosition((glutGet(GLUT\_SCREEN\_WIDTH)-1100)/2,(glutGet(GLUT\_SCREEN\_HEIGHT)-600)/2);  //glutInitWindowPosition();  glutCreateWindow("Lab Task - 4");  //this line must be below of glutCreateWindow();  gluOrtho2D(0, 15, -3, 10);  glutDisplayFunc(display);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |