**Lab Taks-5**

|  |
| --- |
| **Question-1**  Create an animation using two box that will move in the opposite direction. |
| **Graph Plot (Picture)-**  **[Not needed]** |
| **Code-**  #include <windows.h>  #include<GL/gl.h>  #include <GL/glut.h>  #include <math.h>  float \_move1 = 10.0f;  float \_move2 = 10.0f;  void movement()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glMatrixMode(GL\_MODELVIEW);  glPushMatrix();  glTranslatef(\_move1, 0.0f, 0.0f);  glBegin(GL\_POLYGON);  glColor3f(0,1,0);  glVertex2d(-50,60);  glVertex2d(50,60);  glVertex2d(50,20);  glVertex2d(-50,20);  glEnd();  glLineWidth(5);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3f(1,1,1);  glVertex2d(-50,60);  glVertex2d(50,60);  glVertex2d(50,20);  glVertex2d(-50,20);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glPushMatrix();  glTranslatef(\_move2, 0.0f, 0.0f);  glBegin(GL\_POLYGON);  glColor3ub(252, 143, 0);  glVertex2d(-50,-20);  glVertex2d(50,-20);  glVertex2d(50,-60);  glVertex2d(-50,-60);  glEnd();  glLineWidth(5);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3f(1,1,1);  glVertex2d(-50,-20);  glVertex2d(50,-20);  glVertex2d(50,-60);  glVertex2d(-50,-60);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutSwapBuffers();  }  void display()  {  movement();  glFlush();  }  void update1(int value)  {  \_move1 += 7.5;  if(\_move1 > 250)  {  \_move1 = -250.0;  }  glutPostRedisplay();  glutTimerFunc(100, update1, 0);  }  void update2(int value)  {  \_move2 -= 7.5;  if(\_move2 < -250)  {  \_move2 = 250.0;  }  glutPostRedisplay();  glutTimerFunc(100, update2, 0);  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);  glutInitWindowSize(1080, 720);  glutCreateWindow("lab task 5 [22-47226-1]");  glutDisplayFunc(display);  gluOrtho2D(-200,200,-150,150);  glutTimerFunc(100, update1, 0);  glutTimerFunc(100, update2, 0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-2**  Design a car which will have rotating wheels. |
| **Graph Plot (Picture)-**  **[Not needed]** |
| **Code-**  #include <windows.h>  #include<GL/gl.h>  #include <GL/glut.h>  #include <math.h>  float \_angle1 = 0.0f;  void Circle(float radius, float xc, float yc, float r, float g, float b)  {  glLineWidth(7.5);  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 CircleBorder(float radius, float xc, float yc, float width)  {  glLineWidth(width);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3ub(0,0,0);  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();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  }  void frontWheels()  {  glBegin(GL\_POLYGON);  glColor3ub(0,0,0);  glVertex2d(40,5);  glVertex2d(120,5);  glVertex2d(120,-20);  glVertex2d(40,-20);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,0,0);  glVertex2d(25,35);  glVertex2d(95,35);  glVertex2d(95,5);  glVertex2d(60,5);  glVertex2d(45,-25);  glVertex2d(40,-25);  glVertex2d(25,-25);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,0,0);  glVertex2d(95,35);  glVertex2d(130,35);  glVertex2d(130,-25);  glVertex2d(115,-25);  glVertex2d(99,5);  glVertex2d(65,5);  glVertex2d(65,35);  glEnd();  glBegin(GL\_POLYGON); //head light  glColor3ub(195, 209, 0);  glVertex2d(130,23);  glVertex2d(138,23);  glVertex2d(145,13);  glVertex2d(130,13);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //head light  glColor3ub(0, 0, 0);  glVertex2d(130,23);  glVertex2d(138,23);  glVertex2d(145,13);  glVertex2d(130,13);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPushMatrix();  glTranslatef(79.5,-25,0.0f);  glRotatef(\_angle1, 0.0f, 0.0f,-1.0f);  glTranslatef(-79.5,25,0.0f);  Circle(28, 79.5,-25, 61, 61, 54);  Circle(15, 79.5,-25, 0,0,255);  Circle(2, 79.5,-20, 255,0,0);  Circle(2, 79.5,-30, 255,0,0);  Circle(2, 74.5,-25, 255,0,0);  Circle(2, 84.5,-25, 255,0,0);  CircleBorder(28, 79.5,-25, 5);  CircleBorder(15, 79.5,-25, 5);  CircleBorder(2, 79.5,-20, 2);  CircleBorder(2, 79.5,-30, 2);  CircleBorder(2, 74.5,-25, 2);  CircleBorder(2, 84.5,-25, 2);  glPopMatrix();  }  void backWheels()  {  glBegin(GL\_POLYGON);  glColor3ub(0,0,0);  glVertex2d(40-200,5);  glVertex2d(120-200,5);  glVertex2d(120-200,-20);  glVertex2d(40-200,-20);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,0,0);  glVertex2d(40-200,35);  glVertex2d(95-200,35);  glVertex2d(95-200,5);  glVertex2d(60-200,5);  glVertex2d(45-200,-25);  glVertex2d(40-200,-25);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,0,0);  glVertex2d(95-200,35);  glVertex2d(130-200,35);  glVertex2d(130-200,-25);  glVertex2d(115-200,-25);  glVertex2d(99-200,5);  glVertex2d(65-200,5);  glVertex2d(65-200,35);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glLineWidth(2.5);  glBegin(GL\_POLYGON);  glColor3f(0,0,0);  glVertex2d(85-200,25);  glVertex2d(95-200,25);  glVertex2d(95-200,15);  glVertex2d(85-200,15);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //tail light  glColor3ub(195, 209, 0);  glVertex2d(40-200,23);  glVertex2d(55-200,23);  glVertex2d(55-200,13);  glVertex2d(40-200,13);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //tail light border  glColor3ub(0, 0, 0);  glVertex2d(40-200,23);  glVertex2d(55-200,23);  glVertex2d(55-200,13);  glVertex2d(40-200,13);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPushMatrix();  glTranslatef(-120.5,-25,0.0f);  glRotatef(\_angle1, 0.0f, 0.0f,-1.0f);  glTranslatef(120.5,25,0.0f);  Circle(28, -120.5,-25, 61, 61, 54);  Circle(15, -120.5,-25, 0,0,255);  Circle(2, -120.5,-20, 255,0,0);  Circle(2, -120.5,-30, 255,0,0);  Circle(2, -115.5,-25, 255,0,0);  Circle(2, -125.5,-25, 255,0,0);  CircleBorder(28, -120.5,-25, 5);  CircleBorder(15, -120.5,-25, 5);  CircleBorder(2, -120.5,-20, 2);  CircleBorder(2, -120.5,-30, 2);  CircleBorder(2, -115.5,-25, 2);  CircleBorder(2, -125.5,-25, 2);  glPopMatrix();  }  void car()  {  glLineWidth(4);  glMatrixMode(GL\_MODELVIEW);  glBegin(GL\_POLYGON); //upper portion  glColor3f(1,0,0);  glVertex2d(-110,35);  glVertex2d(-90,75);  glVertex2d(10,75);  glVertex2d(40,35);  glEnd();  glBegin(GL\_POLYGON); //windscreen  glColor3f(0,0,1);  glVertex2d(9,70);  glVertex2d(14,70);  glVertex2d(38,38);  glVertex2d(31,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //windscreen border  glColor3f(0,0,0);  glVertex2d(9,70);  glVertex2d(14,70);  glVertex2d(38,38);  glVertex2d(31,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //driver window  glColor3ub(14, 30, 138);  glVertex2d(-55,69);  glVertex2d(3,69);  glVertex2d(25,38);  glVertex2d(-55,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //driver window border  glColor3ub(0, 0, 0);  glVertex2d(-55,69);  glVertex2d(3,69);  glVertex2d(25,38);  glVertex2d(-55,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON);  glColor3ub(14, 30, 138);  glVertex2d(-85,69);  glVertex2d(-63,69);  glVertex2d(-63,38);  glVertex2d(-100,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-85,69);  glVertex2d(-63,69);  glVertex2d(-63,38);  glVertex2d(-100,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3f(0,0,0);  glVertex2d(-110,35);  glVertex2d(-90,75);  glVertex2d(10,75);  glVertex2d(40,35);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON);  glColor3ub(255,0,0);  glVertex2d(130,35);  glVertex2d(165,-15);  glVertex2d(165,-25);  glVertex2d(140,-25);  glVertex2d(130,-25);  glEnd();  frontWheels();  backWheels();  glLineWidth(3);  glBegin(GL\_POLYGON); //spoiler placement  glColor3f(0,0,0);  glVertex2d(-155,45);  glVertex2d(-148,45);  glVertex2d(-148,35);  glVertex2d(-155,35);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //spoiler placement  glColor3f(0,0,0);  glVertex2d(-155,45);  glVertex2d(-148,45);  glVertex2d(-148,35);  glVertex2d(-155,35);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //spoiler bar  glColor3ub(255,0,0);  glVertex2d(-165,45);  glVertex2d(-145,45);  glVertex2d(-145,40);  glVertex2d(-165,40);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //spoiler bar  glColor3f(0,0,0);  glVertex2d(-165,45);  glVertex2d(-145,45);  glVertex2d(-145,40);  glVertex2d(-165,40);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //spoiler fender  glColor3ub(0,0,150);  glVertex2d(-165,45);  glVertex2d(-165,50);  glVertex2d(-158,45);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //spoiler fender  glColor3f(0,0,0);  glVertex2d(-165,45);  glVertex2d(-165,50);  glVertex2d(-158,45);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glLineWidth(3);  glBegin(GL\_POLYGON); //middle body  glColor3ub(255, 0, 0);  glVertex2d(-70,35);  glVertex2d(25,35);  glVertex2d(25,-25);  glVertex2d(-70,-25);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //middle body  glColor3ub(0, 0, 0);  glVertex2d(-70,35);  glVertex2d(25,35);  glVertex2d(25,-25);  glVertex2d(-70,-25);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //door  glColor3ub(130, 22, 0);  glVertex2d(-55,30);  glVertex2d(20,30);  glVertex2d(20,-20);  glVertex2d(-55,-20);  glEnd();  glBegin(GL\_POLYGON); //door handle  glColor3ub(0, 0, 255);  glVertex2d(-45,15);  glVertex2d(-30,15);  glVertex2d(-30,11);  glVertex2d(-45,11);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //door  glColor3ub(0, 0, 0);  glVertex2d(-45,15);  glVertex2d(-30,15);  glVertex2d(-30,11);  glVertex2d(-45,11);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //door  glColor3ub(0, 0, 0);  glVertex2d(-55,30);  glVertex2d(20,30);  glVertex2d(20,-20);  glVertex2d(-55,-20);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3f(0,0,0);  glVertex2d(-160,35);  glVertex2d(130,35);  glVertex2d(165,-15);  glVertex2d(165,-25);  glVertex2d(115,-25);  glVertex2d(99,5);  glVertex2d(60,5);  glVertex2d(45,-25);  glVertex2d(115-200,-25);  glVertex2d(99-200,5);  glVertex2d(60-200,5);  glVertex2d(45-200,-25);  glVertex2d(40-200,-25);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //rearview mirror  glColor3ub(255, 0, 0);  glVertex2d(10,23+18);  glVertex2d(25,23+18);  glVertex2d(25,13+18);  glVertex2d(10,13+18);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //rearview mirror  glColor3ub(0, 0, 0);  glVertex2d(10,23+18);  glVertex2d(25,23+18);  glVertex2d(25,13+18);  glVertex2d(10,13+18);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glutSwapBuffers();  }  void update(int value) {  \_angle1+=5.0f;  if(\_angle1 > 360.0)  {  \_angle1-=360;  }  glutPostRedisplay();  glutTimerFunc(10, update, 0);  }  void display()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.03f, 0.37f, 0.071f, 1.0f);  car();  glutSwapBuffers();  glFlush();  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(1000, 1000);  glutCreateWindow("lab task 5 [22-47226-1]");  glutDisplayFunc(display);  gluOrtho2D(-200,200,-200,200);  glutTimerFunc(20, update, 0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-3**  Now move your car of question-2 from left to right in a loop. |
| **Graph Plot (Picture)-**  **[Not needed]** |
| **Code-**  #include <windows.h>  #include<GL/gl.h>  #include <GL/glut.h>  #include <math.h>  float \_move1 = 10.0f;  float \_angle1 = 0.0f;  void Circle(float radius, float xc, float yc, float r, float g, float b)  {  glLineWidth(7.5);  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 CircleBorder(float radius, float xc, float yc, float width)  {  glLineWidth(width);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3ub(0,0,0);  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();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  }  void frontWheels()  {  glPushMatrix();  glBegin(GL\_POLYGON);  glColor3ub(0,0,0);  glVertex2d(40,5);  glVertex2d(120,5);  glVertex2d(120,-20);  glVertex2d(40,-20);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,0,0);  glVertex2d(25,35);  glVertex2d(95,35);  glVertex2d(95,5);  glVertex2d(60,5);  glVertex2d(45,-25);  glVertex2d(40,-25);  glVertex2d(25,-25);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,0,0);  glVertex2d(95,35);  glVertex2d(130,35);  glVertex2d(130,-25);  glVertex2d(115,-25);  glVertex2d(99,5);  glVertex2d(65,5);  glVertex2d(65,35);  glEnd();  glBegin(GL\_POLYGON); //head light  glColor3ub(195, 209, 0);  glVertex2d(130,23);  glVertex2d(138,23);  glVertex2d(145,13);  glVertex2d(130,13);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //head light  glColor3ub(0, 0, 0);  glVertex2d(130,23);  glVertex2d(138,23);  glVertex2d(145,13);  glVertex2d(130,13);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glPushMatrix();  glTranslatef(79.5,-25,0.0f);  glRotatef(\_angle1, 0.0f, 0.0f,-1.0f);  glTranslatef(-79.5,25,0.0f);  Circle(28, 79.5,-25, 61, 61, 54);  Circle(15, 79.5,-25, 0,0,255);  Circle(2, 79.5,-20, 255,0,0);  Circle(2, 79.5,-30, 255,0,0);  Circle(2, 74.5,-25, 255,0,0);  Circle(2, 84.5,-25, 255,0,0);  CircleBorder(28, 79.5,-25, 5);  CircleBorder(15, 79.5,-25, 5);  CircleBorder(2, 79.5,-20, 2);  CircleBorder(2, 79.5,-30, 2);  CircleBorder(2, 74.5,-25, 2);  CircleBorder(2, 84.5,-25, 2);  glPopMatrix();  }  void backWheels()  {  glBegin(GL\_POLYGON);  glColor3ub(0,0,0);  glVertex2d(40-200,5);  glVertex2d(120-200,5);  glVertex2d(120-200,-20);  glVertex2d(40-200,-20);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,0,0);  glVertex2d(40-200,35);  glVertex2d(95-200,35);  glVertex2d(95-200,5);  glVertex2d(60-200,5);  glVertex2d(45-200,-25);  glVertex2d(40-200,-25);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,0,0);  glVertex2d(95-200,35);  glVertex2d(130-200,35);  glVertex2d(130-200,-25);  glVertex2d(115-200,-25);  glVertex2d(99-200,5);  glVertex2d(65-200,5);  glVertex2d(65-200,35);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glLineWidth(2.5);  glBegin(GL\_POLYGON);  glColor3f(0,0,0);  glVertex2d(85-200,25);  glVertex2d(95-200,25);  glVertex2d(95-200,15);  glVertex2d(85-200,15);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //tail light  glColor3ub(195, 209, 0);  glVertex2d(40-200,23);  glVertex2d(55-200,23);  glVertex2d(55-200,13);  glVertex2d(40-200,13);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //tail light border  glColor3ub(0, 0, 0);  glVertex2d(40-200,23);  glVertex2d(55-200,23);  glVertex2d(55-200,13);  glVertex2d(40-200,13);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPushMatrix();  glTranslatef(-120.5,-25,0.0f);  glRotatef(\_angle1, 0.0f, 0.0f,-1.0f);  glTranslatef(120.5,25,0.0f);  Circle(28, -120.5,-25, 61, 61, 54);  Circle(15, -120.5,-25, 0,0,255);  Circle(2, -120.5,-20, 255,0,0);  Circle(2, -120.5,-30, 255,0,0);  Circle(2, -115.5,-25, 255,0,0);  Circle(2, -125.5,-25, 255,0,0);  CircleBorder(28, -120.5,-25, 5);  CircleBorder(15, -120.5,-25, 5);  CircleBorder(2, -120.5,-20, 2);  CircleBorder(2, -120.5,-30, 2);  CircleBorder(2, -115.5,-25, 2);  CircleBorder(2, -125.5,-25, 2);  glPopMatrix();  }  void car()  {  glLineWidth(4);  glMatrixMode(GL\_MODELVIEW);  glPushMatrix();  glBegin(GL\_POLYGON); //upper portion  glColor3f(1,0,0);  glVertex2d(-110,35);  glVertex2d(-90,75);  glVertex2d(10,75);  glVertex2d(40,35);  glEnd();  glBegin(GL\_POLYGON); //windscreen  glColor3f(0,0,1);  glVertex2d(9,70);  glVertex2d(14,70);  glVertex2d(38,38);  glVertex2d(31,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //windscreen border  glColor3f(0,0,0);  glVertex2d(9,70);  glVertex2d(14,70);  glVertex2d(38,38);  glVertex2d(31,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //driver window  glColor3ub(14, 30, 138);  glVertex2d(-55,69);  glVertex2d(3,69);  glVertex2d(25,38);  glVertex2d(-55,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //driver window border  glColor3ub(0, 0, 0);  glVertex2d(-55,69);  glVertex2d(3,69);  glVertex2d(25,38);  glVertex2d(-55,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON);  glColor3ub(14, 30, 138);  glVertex2d(-85,69);  glVertex2d(-63,69);  glVertex2d(-63,38);  glVertex2d(-100,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-85,69);  glVertex2d(-63,69);  glVertex2d(-63,38);  glVertex2d(-100,38);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3f(0,0,0);  glVertex2d(-110,35);  glVertex2d(-90,75);  glVertex2d(10,75);  glVertex2d(40,35);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON);  glColor3ub(255,0,0);  glVertex2d(130,35);  glVertex2d(165,-15);  glVertex2d(165,-25);  glVertex2d(140,-25);  glVertex2d(130,-25);  glEnd();  frontWheels();  backWheels();  glLineWidth(3);  glBegin(GL\_POLYGON); //spoiler placement  glColor3f(0,0,0);  glVertex2d(-155,45);  glVertex2d(-148,45);  glVertex2d(-148,35);  glVertex2d(-155,35);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //spoiler placement  glColor3f(0,0,0);  glVertex2d(-155,45);  glVertex2d(-148,45);  glVertex2d(-148,35);  glVertex2d(-155,35);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //spoiler bar  glColor3ub(255,0,0);  glVertex2d(-165,45);  glVertex2d(-145,45);  glVertex2d(-145,40);  glVertex2d(-165,40);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //spoiler bar  glColor3f(0,0,0);  glVertex2d(-165,45);  glVertex2d(-145,45);  glVertex2d(-145,40);  glVertex2d(-165,40);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //spoiler fender  glColor3ub(0,0,150);  glVertex2d(-165,45);  glVertex2d(-165,50);  glVertex2d(-158,45);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //spoiler fender  glColor3f(0,0,0);  glVertex2d(-165,45);  glVertex2d(-165,50);  glVertex2d(-158,45);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glLineWidth(3);  glBegin(GL\_POLYGON); //middle body  glColor3ub(255, 0, 0);  glVertex2d(-70,35);  glVertex2d(25,35);  glVertex2d(25,-25);  glVertex2d(-70,-25);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //middle body  glColor3ub(0, 0, 0);  glVertex2d(-70,35);  glVertex2d(25,35);  glVertex2d(25,-25);  glVertex2d(-70,-25);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //door  glColor3ub(130, 22, 0);  glVertex2d(-55,30);  glVertex2d(20,30);  glVertex2d(20,-20);  glVertex2d(-55,-20);  glEnd();  glBegin(GL\_POLYGON); //door handle  glColor3ub(0, 0, 255);  glVertex2d(-45,15);  glVertex2d(-30,15);  glVertex2d(-30,11);  glVertex2d(-45,11);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //door  glColor3ub(0, 0, 0);  glVertex2d(-45,15);  glVertex2d(-30,15);  glVertex2d(-30,11);  glVertex2d(-45,11);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //door  glColor3ub(0, 0, 0);  glVertex2d(-55,30);  glVertex2d(20,30);  glVertex2d(20,-20);  glVertex2d(-55,-20);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3f(0,0,0);  glVertex2d(-160,35);  glVertex2d(130,35);  glVertex2d(165,-15);  glVertex2d(165,-25);  glVertex2d(115,-25);  glVertex2d(99,5);  glVertex2d(60,5);  glVertex2d(45,-25);  glVertex2d(115-200,-25);  glVertex2d(99-200,5);  glVertex2d(60-200,5);  glVertex2d(45-200,-25);  glVertex2d(40-200,-25);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //rearview mirror  glColor3ub(255, 0, 0);  glVertex2d(10,23+18);  glVertex2d(25,23+18);  glVertex2d(25,13+18);  glVertex2d(10,13+18);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //rearview mirror  glColor3ub(0, 0, 0);  glVertex2d(10,23+18);  glVertex2d(25,23+18);  glVertex2d(25,13+18);  glVertex2d(10,13+18);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutSwapBuffers();  }  void update(int value) {  \_angle1+=5.0f;  if(\_angle1 > 360.0)  {  \_angle1-=360;  }  glutPostRedisplay();  glutTimerFunc(10, update, 0);  }  void update1(int value)  {  \_move1 += 5.5;  if(\_move1 > 380)  {  \_move1 = -360.0;  }  glutPostRedisplay();  glutTimerFunc(100, update1, 0);  }  void road()  {  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-200,20);  glVertex2d(200,20);  glVertex2d(200,-70);  glVertex2d(-200,-70);  glEnd();  }  void display()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.03f, 0.37f, 0.071f, 1.0f);  road();  glMatrixMode(GL\_MODELVIEW);  glPushMatrix();  glTranslatef(\_move1,0,0.0f);  car();  glPopMatrix();  glutSwapBuffers();  glFlush();  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(1000, 1000);  glutCreateWindow("lab task 5 [22-47226-1]");  glutDisplayFunc(display);  gluOrtho2D(-200,200,-200,200);  glutTimerFunc(20, update, 0);  glutTimerFunc(5, update1, 0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-4**  Design a windmill with rotating blades |
| **Graph Plot (Picture)-**  **[Not needed]** |
| **Code-**  #include <windows.h>  #include<GL/gl.h>  #include <GL/glut.h>  #include <math.h>  float \_move1 = 10.0f;  float \_angle1 = 0.0f;  void Circle(float radius, float xc, float yc, float r, float g, float b)  {  glLineWidth(7.5);  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 CircleBorder(float radius, float xc, float yc, float width)  {  glLineWidth(width);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3ub(0,0,0);  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();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  }  void update(int value) {  \_angle1+=2.0f;  if(\_angle1 > 360.0)  {  \_angle1-=360;  }  glutPostRedisplay();  glutTimerFunc(10, update, 0);  }  void tower()  {  glBegin(GL\_POLYGON); //tower  glColor3ub(184, 82, 18);  glVertex2d(-20,-60);  glVertex2d(-20,50);  glVertex2d(20,50);  glVertex2d(20,-60);  glEnd();  glLineWidth(3);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //tower border  glColor3ub(0, 0, 0);  glVertex2d(-20,-60);  glVertex2d(-20,50);  glVertex2d(20,50);  glVertex2d(20,-60);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //tower head  glColor3ub(140, 38, 4);  glVertex2d(-30,50);  glVertex2d(0,80);  glVertex2d(30,50);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //tower head border  glColor3ub(0, 0, 0);  glVertex2d(-30,50);  glVertex2d(0,80);  glVertex2d(30,50);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //tower door  glColor3ub(0, 40, 30);  glVertex2d(-10,-60);  glVertex2d(-10,-30);  glVertex2d(10,-30);  glVertex2d(10,-60);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //tower door border  glColor3ub(0, 0, 0);  glVertex2d(-10,-60);  glVertex2d(-10,-30);  glVertex2d(10,-30);  glVertex2d(10,-60);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //tower window  glColor3ub(158, 172, 176);  glVertex2d(-5,-60+50);  glVertex2d(-5,-50+50);  glVertex2d(5,-50+50);  glVertex2d(5,-60+50);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //tower window 1 border  glColor3ub(0, 0, 0);  glVertex2d(-5,-60+50);  glVertex2d(-5,-50+50);  glVertex2d(5,-50+50);  glVertex2d(5,-60+50);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //tower window  glColor3ub(158, 172, 176);  glVertex2d(-5,-60+70);  glVertex2d(-5,-50+70);  glVertex2d(5,-50+70);  glVertex2d(5,-60+70);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //tower window 2 border  glColor3ub(0, 0, 0);  glVertex2d(-5,-60+70);  glVertex2d(-5,-50+70);  glVertex2d(5,-50+70);  glVertex2d(5,-60+70);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glBegin(GL\_POLYGON); //tower window  glColor3ub(158, 172, 176);  glVertex2d(-5,-60+90);  glVertex2d(-5,-50+90);  glVertex2d(5,-50+90);  glVertex2d(5,-60+90);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON); //tower window 3 border  glColor3ub(0, 0, 0);  glVertex2d(-5,-60+90);  glVertex2d(-5,-50+90);  glVertex2d(5,-50+90);  glVertex2d(5,-60+90);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  }  void blades()  {  glMatrixMode(GL\_MODELVIEW);  glPushMatrix();  glTranslatef(0,63,0.0f);  glRotatef(\_angle1, 0.0f, 0.0f,-1.0f);  glTranslatef(-0,-63,0.0f);  glBegin(GL\_POLYGON); //horizontal line  glColor3ub(10, 0, 0);  glVertex2d(-80,0+60);  glVertex2d(80,0+60);  glVertex2d(80,6+60);  glVertex2d(-80,6+60);  glEnd();  glBegin(GL\_POLYGON); //red sails  glColor3ub(200, 0, 0);  glVertex2d(-80,6+60);  glVertex2d(-80,26+60);  glVertex2d(-20,26+60);  glVertex2d(-20,6+60);  glEnd();  glBegin(GL\_POLYGON); //blue sails  glColor3ub(0, 0, 200);  glVertex2d(80,0+60);  glVertex2d(80,-20+60);  glVertex2d(20,-20+60);  glVertex2d(20,0+60);  glEnd();  glBegin(GL\_POLYGON); //vertical line  glColor3ub(10, 0, 0);  glVertex2d(0-3,-80+60);  glVertex2d(0-3,80+60);  glVertex2d(6-3,80+60);  glVertex2d(6-3,-80+60);  glEnd();  glBegin(GL\_POLYGON); //yellow sails  glColor3ub(251, 255, 0);  glVertex2d(6-3,80+60);  glVertex2d(26-3,80+60);  glVertex2d(26-3,20+60);  glVertex2d(6-3,20+60);  glEnd();  glBegin(GL\_POLYGON); //green sails  glColor3ub(0, 200, 0);  glVertex2d(0-3,-80+60);  glVertex2d(-20-3,-80+60);  glVertex2d(-20-3,-20+60);  glVertex2d(0-3,-20+60);  glEnd();  Circle(5,0,63,118, 13, 252);  CircleBorder(5,0,63,2);  glPopMatrix();  }  void display()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.7f, 0.7f, 0.65f, 1.0f);  tower();  blades();  glutSwapBuffers();  glFlush();  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(1000, 1000);  glutCreateWindow("lab task 5 [22-47226-1]");  glutDisplayFunc(display);  gluOrtho2D(-150,150,-150,150);  glutTimerFunc(20, update, 0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |