9) Write a program to model a car like figure using display lists and move a car from one end of the screen to other end. User is able to control the speed with mouse.

```
#include<GL/glut.h>
                        #include<math.h>
                        #include<stdio.h>
                        #define CAR 1
                        #define WHEEL 2
                        float s = 1;
                        void carlist() {


                                glNewList(CAR, GL_COMPILE);
                                glColor3f(1, 1, 1);
                                glBegin(GL_POLYGON);
                                glVertex3f(0, 25, 0);
                                glVertex3f(90, 25, 0);
                                glVertex3f(90, 55, 0);
                                glVertex3f(80, 55, 0);
                                glVertex3f(20, 75, 0);
                                glVertex3f(0, 55, 0);
                                glEnd();
                                glEndList();


                        }
                        void wheellist() {
                                glNewList(WHEEL, GL_COMPILE_AND_EXECUTE);
                                glColor3f(0, 1, 1);
                                glutSolidSphere(10, 25, 25);
                                glEndList();
                        }
                        void mykeyboard(unsigned char key, int x, int y) {
                                switch (key) {
                                case 't': glutPostRedisplay();
                                        break;
                                case 'q': exit(0);
                                default: break;


                                }
                        }
```

```
void myInit() {
	glClearColor(0, 0, 0, 0);
	glOrtho(0, 600, 0, 600, 0, 600);


}
void draw_wheel() {
	glColor3f(0, 1, 1);
	glutSolidSphere(10, 25, 25);


}


void moveCar(float s) {
	glTranslatef(s, 0.0, 0.0);
	glCallList(CAR);
	glPushMatrix();
	glTranslatef(25, 25, 0.0);	//move to first wheel
position
	//draw_wheel();
	glCallList(WHEEL);
	glPopMatrix();
	glPushMatrix();
	glTranslatef(75, 25, 0.0);	//move to 2nd wheel
position
	////draw_wheel();
	glCallList(WHEEL);
	glPopMatrix();
	glFlush();
}
void myDisp() {
	glClear(GL_COLOR_BUFFER_BIT);
	carlist();
	moveCar(s);
	wheellist();


}
void mouse(int btn, int state, int x, int y) {
	if (btn == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
		s += 5;
		myDisp();
	}
	else if (btn == GLUT_RIGHT_BUTTON && state == GLUT_DOWN) {
		s += 2;
```

```
                myDisp();
        }
}


int main(int argc, char* argv[]) {
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowSize(600, 500);
        glutInitWindowPosition(100, 100);
        glutCreateWindow("car");
        myInit();
        glutDisplayFunc(myDisp);
        glutMouseFunc(mouse);
        glutKeyboardFunc(mykeyboard);
        glutMainLoop();
}
```