

1)  
ambient(0.0,0.0,0.0)  
diffuse(1.0,1.0,1.0)  
specular(1.0,1.0,1.0)

**Bài thực hành ánh sáng** (Tài liệu tham khảo: Chương 5, OGLRB)

## I. Chạy chương trình light.c

1, Chương trình sử dụng duy nhất một nguồn sáng trắng; vị trí của nó được xác định bằng lời gọi `glLightfv()`. Ví dụ này sử dụng màu sắc mặc định cho `GL_LIGHT0`, đó là màu trắng. Phân biệt vùng nào trong khối cầu là ánh sáng `ambient`, `diffuse`, `specular`. Cho biết giá trị mặc định của 3 thuộc tính này.

2, Bỏ tạm thời dòng lệnh `glEnable(GL_LIGHTING)`; và quan sát kết quả chạy chương trình. Giải thích kết quả? *Hiện thị mặt cầu đặc màu trắng do chưa có lệnh kích hoạt nguồn sáng `glEnable(...)`*

3, Giải thích lệnh `glEnable (GL_LIGHT0)`; *Bật nguồn sáng sso 0 mặc định là màu trắng*

4 Trong cú pháp lệnh tạo nguồn sáng: `void glLight{if}v(GLenum light, GLenum pname, TYPEparam)`; Đối số `light` nhận những giá trị nào, cho một ví dụ về tạo một nguồn sáng, giải thích? *Gồm 3 giá trị là ánh sáng có thuộc tính dạng được can xác định, thuộc tính, và giá trị mong muốn cho thuộc tính đó*

## II. Viết chương trình.

Bài 1. Viết chương trình hiển thị khối cầu được chiếu sáng có màu xanh da trời.

Bài 2: a. Viết chương trình hiển thị hai ảm trà được chiếu sáng, một ảm màu vàng (yellow), một ảm màu tím (magenta)

b. Khi nhấn phím 'a', ảm trà vàng quanh quanh chính nó.

Bài 3.

Tham khảo chương trình chuyển động của quả bóng trên thanh trượt như sau, hãy bổ sung các lệnh để quả bóng được chiếu sáng màu xanh lá cây, thanh trượt màu đỏ.

```
#include <GL/gl.h>
```

```
#include <GL/glu.h>
```

```
#include <GL/glut.h>
```

```
static int day= 0;
```

```
void display(void){
```

```
    glClear (GL_COLOR_BUFFER_BIT);
```

```
    glPushMatrix();
```

```
        glColor3f(0.0,0.0,0.0);
```

```

        glRotatef(45,0,0,1);
        glScalef(15,0.6,5);
        glutWireCube(1.0);
glPopMatrix();
glPushMatrix();
        glColor3f(1.0,0.0,1.0);
        glTranslatef(5,7,0);
        glTranslatef( (GLfloat) day/40 , (GLfloat) day/40 , 0);
        glRotatef (((GLfloat) day)*(-1), 0, 0, 1);
        glutWireSphere(1, 20, 15);

glPopMatrix();
glutSwapBuffers();}

void reshape (int w, int h){
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluPerspective(60.0, (GLfloat) w/(GLfloat) h, 1.0, 20.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt (0, 0, 20, 0.0, 0.0, 0.0, 0, 1, 0);
}

void keyboard (unsigned char key, int x, int y){
    switch (key) {
    case '.':
        day = (day + 1)/1.1;
        glutPostRedisplay();
        break;

```

```

        case ',':
            day = (day - 1)*1.1;
            glutPostRedisplay();
            break;
        default:
            break;
    }
}

void init(void) {
    glClearColor (1.0, 1.0, 1.0, 1.0);
    glShadeModel (GL_FLAT);}

int main(int argc, char** argv){
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize (800, 800);
    glutInitWindowPosition (0, 0);
    glutCreateWindow (argv[0]);
    init ();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}

```