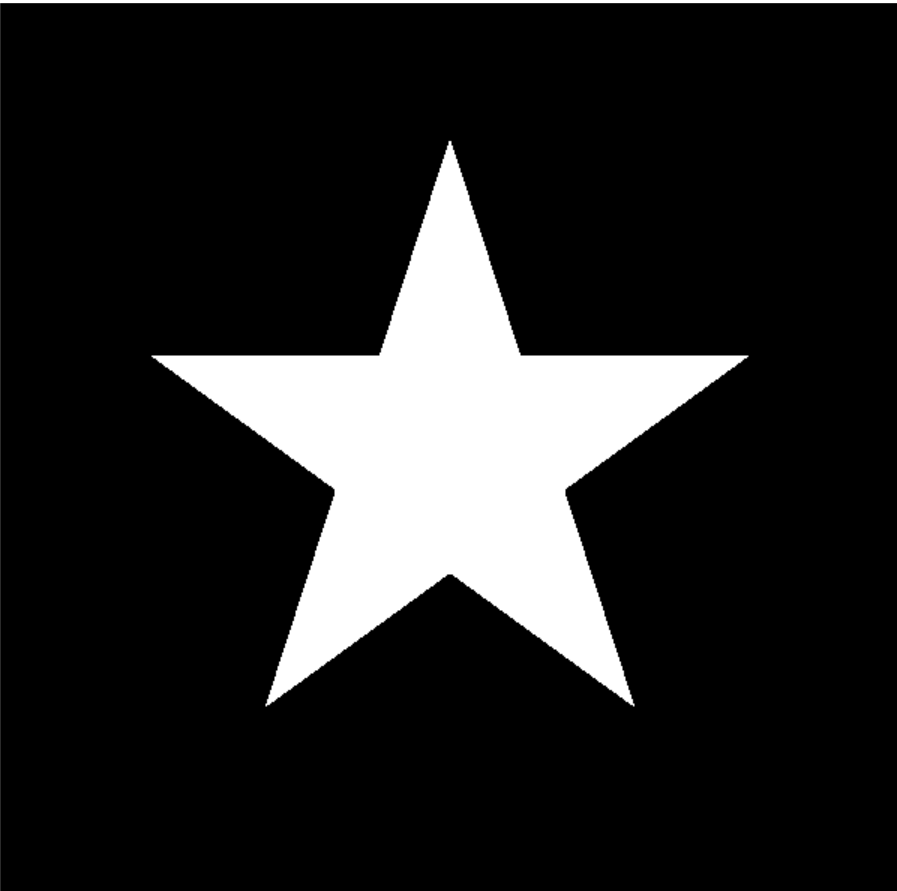
平时作业1

刘骏霖 520021910957

**程序使用说明**

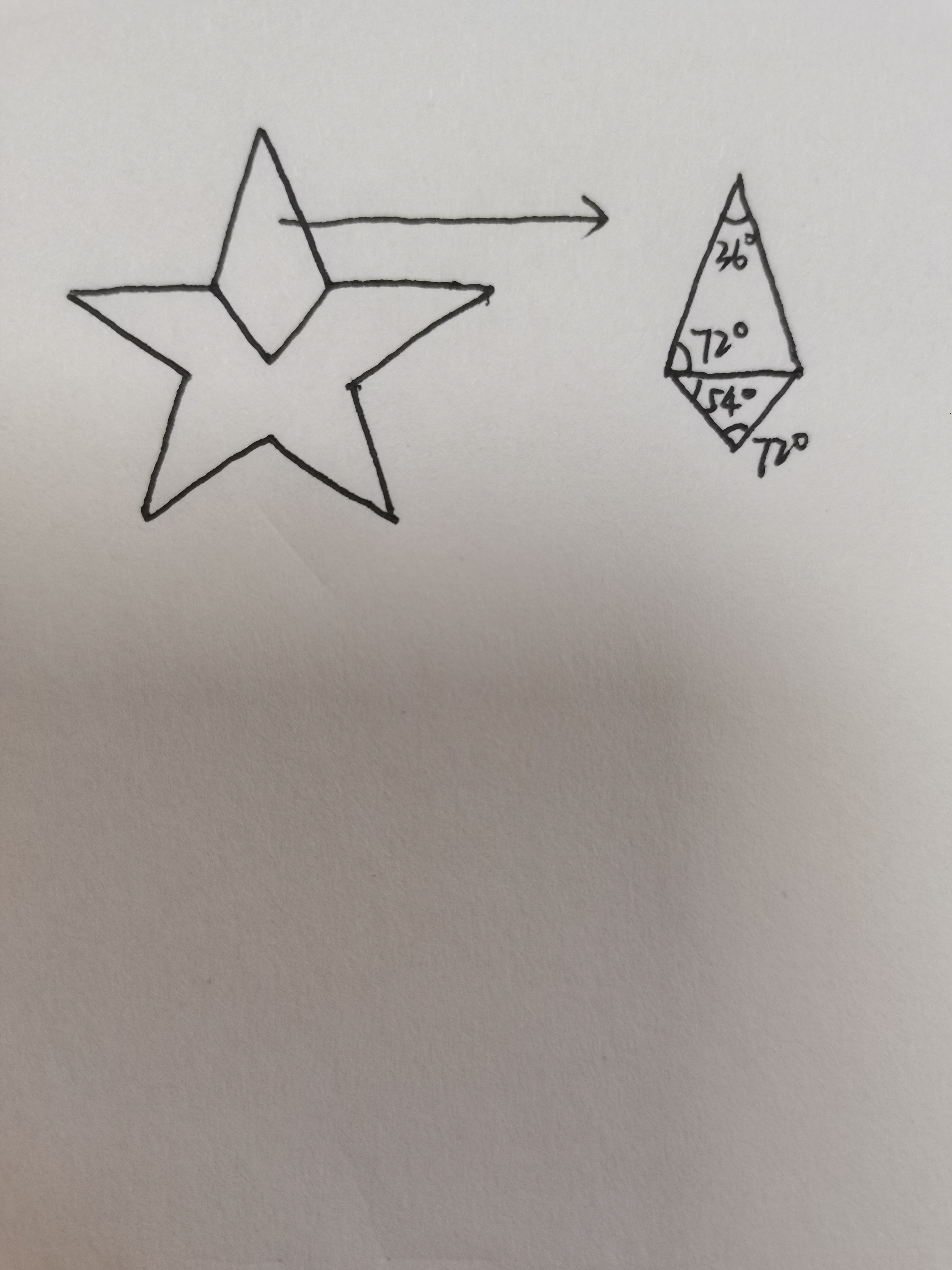
本次作业要实现一个发着光的五角星，根据作业要求，我实现的五角星如下：

首先展示一个黑底白色的五角星，在按下‘c’键后，五角星将上色变得立体，在按下‘s’键后，五角星背后将有一个“小太阳”发光。

1. **展示一个黑底白色的五角星**

我先在纸上绘图得到了五角星的各项数据如图：



然后把一个五角星分成了十个三角形，也就是对应图中每个右边部分分成左右两个三角形，以五角星中心为原点（0,0,0），根据五角星的对称关系，让角度旋转五次次画出了整个五角星。其中用到了绘制三角形的函数glBegin(GL\_TRIANGLES)。

需要注意的是，在绘制过程中虽然我设置了第三维坐标值来提供立体感，但是展示画面上视角没有变动所以第三维实际上没有在图上体现。

1. **给五角星上色来提供立体感**

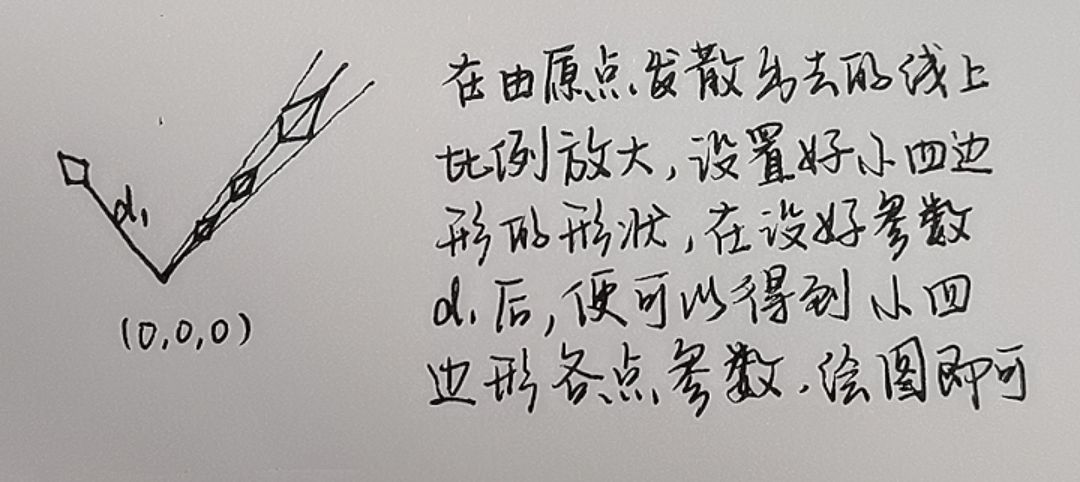
当我们按下‘c’键时，会将五角星重绘一次，在白色五角星的基础上增加了判断，如果当前是‘c’键状态，也就是对应代码中starMode==1的状态，则给五角星上两种不同的红色。它们是（1,0,0）和（0.6,0,0）。同时，在实现中也运用到了glutKeyboardFunc(keyController);的函数来帮助我们处理按键功能。在keyController中，当按下‘c’时，我们将starMode设为1并调用重绘函数glutPostRedisplay()；当按下‘s’时，我们设置好参数开始处理发光的情况。

1. **让五角星背后的太阳发光**

这一部分又分为两个part。

PartA是绘制“太阳”也就是一个圆，由于我目前掌握的opengl还不支持画圆，所以我采用了将圆看成一个正200边形再分割成200个三角形的方法来画圆。“太阳”颜色采用(1,0.5,0)。

PartB是绘制“太阳”发出的光，我们可以看做一圈又一圈的小四边形往外发散的过程，于是我设置好四边形的参数（各店与原点的距离与角度），一共设置了三层小四边形（对应代码中的d[5][5]数组），根据发散前后顺序来设置在某个时间点这个小四边形形是否展示。画小四边形的过程如图：



这里我们需要运用到glutTimerFunc(300, blingTimer, 10)函数来帮助我们每隔300ms刷新一次五角星来实现发散功能，在blingTimer()函数中我设置好shiningType参数（根据发散性质每6一个循环），再在display()函数中根据shiningType参数来绘制小四边形。