中国地质大学（武汉）课程考核结课考试试卷 教务处制 版本：2014.12

试卷类别

A ☑

B ☐

使用学期

年

春☐秋☑

命题人签字

审题人签字

审定人签字

考生学号

考生姓名

所在班级

装 订 线

课程名称：计算机图形学 学时：48

考试时长：120 分钟 卷面总分：100 分

考试方式：闭卷笔试☑ 开卷笔试 ☐ 口试 ☐ 其它____

辅助工具：可用☑ 工具名称：计算器和直尺 不可用☐

注意事项：

（1）请将 考场座位号 写在答题纸左边装订区内；

（2）各题的答案请全部写在答题纸上（写在试卷纸上无效）；

（3）书写答案时请使用蓝色或黑色钢笔、签字笔、水芯笔或圆珠笔（不要使用铅笔）。

试题内容：

一、单项选择题（有10小题，每小题2分，共20分）

1、要准确画出一个多边形的形状，提供的数据应该是（ ）。

A）顶点坐标和颜色 B）顶点坐标和拓扑信息

C）顶点坐标 D）拓扑信息和颜色

2、下列设备中属于图形输出设备的是（ ）。

①空间球 ②LCD ③键盘 ④ LED

⑤打印机 ⑥扫描仪 ⑦绘图仪 ⑧数字化仪

A）①③⑥⑧ B）④⑥⑦⑧

C）②⑤⑥⑦ D）②④⑤⑦

3、在椭圆弧的中点 Bresenham 扫描转换算法中，如果针对的是中心在原点，第一象限的 1/4 段椭圆弧，则其上下部分的分界点是（ ）。

A）椭圆弧上切线斜率为 1 的点

B）椭圆弧与直线 $y=x$ 的交点

C）椭圆弧上法向量 x、y 两个分量相等的点

D）椭圆弧上法向量 x、y 两个分量互为相反数的点

4、以下关于反走样的论述中错误的是（ ）。

A）提高分辨率 B）把像素当作平面区域进行采样

C）采用锥形滤波器进行加权区域采样 D）增强图像的显示亮度

5、使用下列二维图形变换矩阵，将产生变换的结果为（ ）。

$$T=\begin{bmatrix}1 & 0 & 0\\ 0 & 3 & 0\\ 1 & 1 & 1\end{bmatrix}$$

A）图形放大 3 倍

B）沿 Y 坐标轴方向各移动 3 个绘图单位

C）沿 Y 坐标轴方向放大 3 倍，同时沿 X、Y 坐标轴方向各平移 1 个绘图单位

D）图形放大 3 倍，同时沿 X、Y 坐标轴方向各移动 1 个绘图单位

6、三维几何透视投影中，主灭点的个数最多是（　　）。

A）1　　B）2　　C）3　　D）无数个

7、在三维观察中，引入规范化投影空间的目的是（　　）。

A）使观察变换与设备无关　　B）减少存储空间

C）简化投影变换　　D）便于裁剪

8、下列有关平面几何投影的叙述语句中，正确的论述为（　　）。

A）在平面几何投影中，若投影中心移到距离投影面无穷远处，则成为平行投影

B）透视投影变换中，一组平行线投影在与之平行的投影面上，会产生灭点

C）透视投影与平行投影相比，视觉效果更有真实感，而且能真实地反映物体的精确的尺寸和形状

D）在三维空间中的物体进行透视投影变换，可能产生三个或者更多的主灭点。

9、OpenGL 中进行矩阵操作之前，需要指定当前操作的矩阵对象，可以使用函数 glMatrixMode（GLenum mode），mode 取（　　）表示对模型视图矩阵进行操作。

A）GL_PROJECTION　　B）GL_QUADS　　C）GL_MODELVIEW　　D）GL_TEXTURE

10、下列不属于消隐算法的是（　　）。

A）光线投射算法　　B）画家算法

C）编码裁剪算法　　D）深度缓存器算法

二、填空题（有10小题，每小题1分，共10分）

1、计算机中表示带有颜色及形状信息的图和形常用两种方法，包括参数法和（　1　）。

2、阴极射线管（CRT）从结构上主要包括电子枪、（　2　）和荧光屏。

3、光栅扫描显示器中，电子束在扫描线之间的回扫期称为水平回扫，在帧之间的回扫期称为（　3　）。

4、如果某图形系统的显示分辨率为 1280×1024，颜色为 24 位真彩色，则帧缓存最小为（　4　）MB.

5、x-扫描线算法中，每次用一条扫描线进行填充，对一条扫描线填充的过程可分为 4 个步骤：求交、（　5　）、交点配对、区间填色。

6、用于减少或克服在“光栅图形显示器上绘制直线、多边形等连续图形时，由离散量表示连续量引起的失真”的技术叫（　6　）。

7、三维空间点 $p(x, y, z)$的规范化齐次坐标表示是（　7　）

8、平面几何投影分为（　8　）和透视投影两类。

9、用奇偶规则判断点与区域的内外关系的基本思想是：从该点引出任意一条射线，若射线与区域的交点数目为（　9　）时，则点在区域外。

10、在边缘填充算法中引入（　10　）可以减少象素重复访问次数。

三、综合题（有6小题，每小题10分，共60分）

1、使用 DDA 算法进行直线扫描转换，请写出直线段从点 P(1,1)到点 Q(10,4)经过的象素点并给出每步计算步骤。

装

订

线

考生学号

考生姓名

所在班级

2、请简要说明三维观察流程的主要步骤。

3、如图 1 所示，裁剪窗口为正方形，试用 Sutherland-Hodgeman 多边形裁剪算法将裁剪窗口边界按左、下、右、上的顺序依次对如图所示的多边形 ABCDE 进行裁剪（与裁剪线的交点已用数字标出），要求画出每次裁剪对应的图形，并标明输入和输出的顶点序列。

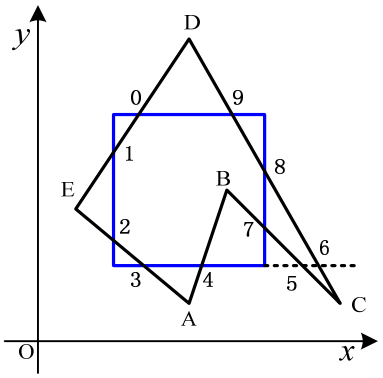


图 1

4、试作出图 2 中三维形体 ABCDE 的三视图（已知平移量均为 1）。要求写清变换过程，并画出生成的三视图。

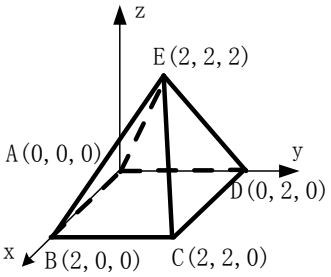


图 2

5、如图 3 所示多边形 ABCDEF，若采用改进的有效边表算法进行填充，在填充时采用“下闭上开”的原则（即删除 y=ymax 的边之后再填充），试写出该多边形的 ET 表和当扫描线 Y=6 时的 AET 表。

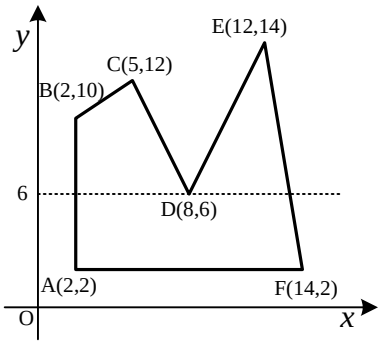


图 3

6、给定四点 P1(1, 3)，P2(4, 6)，P3 (7, 6)，P4(10, 3)，用其作为特征多边形来构造一条三次 Bezier 曲线，写出曲线的参数方程，计算参数分别为 0，1/3，2/3，1 时的值各为多少，并画出曲线。

四、程序题（有1小题，每小题10分，共10分）

如图 4 所示多边形 ABCD，将其绕点 P（3, 3）顺时针旋转 90^0，请写出完整的 OpenGL 程序（部分程序已给出）。

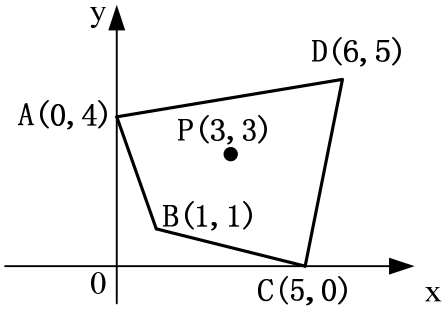


图 4

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

void Display( ){
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0f, 0.0f, 0.0f);

    // 需将此处补充完整……


    glFlush( );
}

void main(int argc, char* argv[ ]){
    glutInit(&argc,  argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutCreateWindow("OpenGL");
    glutDisplayFunc(Display); // 设置当前窗口的显示回掉函数
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-20.0, 20.0, -20.0, 20.0);    // 设置投影面的模型坐标范围
    glutMainLoop(); // 启动主 GLUT事件处理循环
}
```

一、单项选择题（有10小题，每小题2分，共20分）

1-5：BDCDC　　6-10：CDACC

二、填空题（有10小题，每小题1分，共10分）

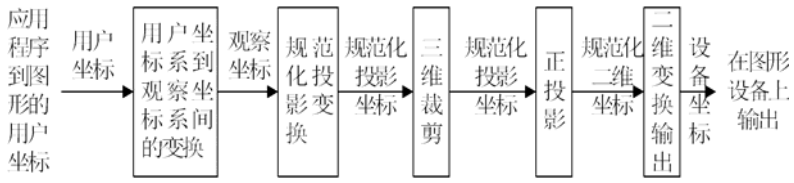
1）点阵法　2）偏转系统　3）垂直回扫　4）3.75　5）排序
6）反走样　7）(x, y, z, 1)　8）平行投影　9）偶数　10）栅栏

三、综合题（有6小题，每小题10分，共60分）

1、答：$k = \frac{y_1 - y_0}{x_1 - x_0} = 1/3$　➔y = x/3+2/3

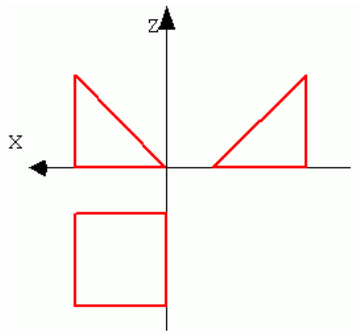
⇨ （1,1），（2,1），（3,2），（4,2），（5,2），
（6,3），（7,3），（8,3），（9,4），（10,4）

2、答：



3、答：　用左边界裁剪：输入：ABCDE；　输出：BCD12A
用下边界裁剪：输入：BCD12A；　输出：56D1234B
用右边界裁剪：输入：56D1234B；　输出：8D1234B7
用上边界裁剪：输入：8D1234B7；　输出：901234B78

4、答：



正视图：

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P' = P \cdot T = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 2 & 2 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 2 & 1 \end{bmatrix}$$

俯视图：

$$T=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}\cdot\begin{bmatrix}1&0&0&0\\0&0&-1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}\cdot\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&-1&1\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&0&-1&0\\0&0&0&0\\0&0&-1&1\end{bmatrix}$$

$$P'=P\cdot T=\begin{bmatrix}0&0&0&1\\2&0&0&1\\2&2&0&1\\0&2&0&1\\2&2&2&1\end{bmatrix}\cdot\begin{bmatrix}1&0&0&0\\0&0&-1&0\\0&0&0&0\\0&0&-1&1\end{bmatrix}=\begin{bmatrix}0&0&-1&1\\2&0&-1&1\\2&0&-3&1\\0&0&-3&1\\2&0&-3&1\end{bmatrix}$$

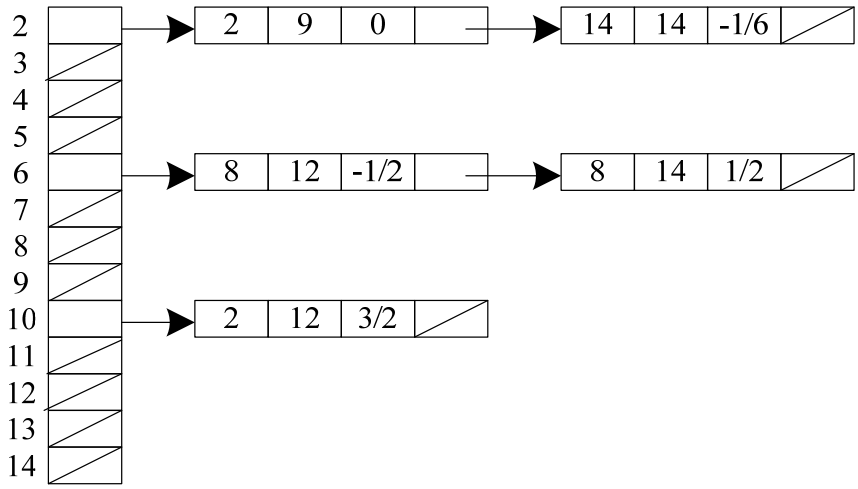
侧视图：

$$T=\begin{bmatrix}0&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\cdot\begin{bmatrix}0&1&0&0\\-1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\cdot\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\-1&0&0&1\end{bmatrix}=\begin{bmatrix}0&0&0&0\\-1&0&0&0\\0&0&1&0\\-1&0&0&1\end{bmatrix}$$

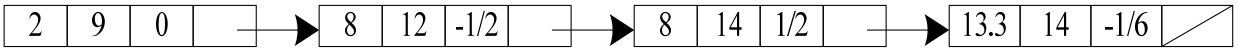
$$P'=P\cdot T=\begin{bmatrix}0&0&0&1\\2&0&0&1\\2&2&0&1\\0&2&0&1\\2&2&2&1\end{bmatrix}\cdot\begin{bmatrix}0&0&0&0\\-1&0&0&0\\0&0&1&0\\-1&0&0&1\end{bmatrix}=\begin{bmatrix}-1&0&0&1\\-1&0&0&1\\-1&0&0&1\\-3&0&0&1\\-3&0&2&1\end{bmatrix}$$

5、答：

边表为：



Y=6 时的有效边表为：



6、答：三次 Bezier 曲线参数方程为：$P(t)=B_{0,3}(t)P_1+B_{1,3}(t)P_2+B_{2,3}(t)P_3+B_{3,3}(t)P_4$

其中：$B_{0,3}(t)=(1-t)^3$

$B_{1,3}(t)=3t(1-t)^2$

$B_{2,3}(t)=3t^2(1-t)$

$B_{3,3}(t)=t^3$

当 t=0 时，P(t) = P1 = [1　3]

当 t=1 时，P(t) = P4 = [10　3]

当 t=1/3 时，P(t)=8/27 P1+4/9 P2+ 2/9 P3 + 1/27 P4 = [4.0　5.0]
当 t=2/3 时，P(t)=1/27 P1+2/9 P2+ 4/9 P3 + 8/27 P4 = [7.0　5.0]
图略

四、程序题（有1小题，每小题10分，共10分）

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

void Display( ){
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0f, 0.0f, 0.0f);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glTranslated(3.0, 3.0, 0.0);
    glRotated(-90.0, 0.0, 0.0, 1.0);
    glTranslated(-3.0, -3.0, 0.0);
    glBegin(GL_POLYGONS);
        glVertex2d(0, 4);
        glVertex2d(1, 1);
        glVertex2d(5, 0);
        glVertex2d(6, 5);
    glEnd();
    glFlush();
}

void main(int argc, char* argv[ ]){
    glutInit(&argc,  argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutCreateWindow("OpenGL");
    glutDisplayFunc(Display); // 设置当前窗口的显示回掉函数
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-20.0, 20.0, -20.0, 20.0);   // 设置投影面的模型坐标范围
    glutMainLoop(); // 启动主 GLUT事件处理循环
}
```