NJFU计算机图形学期末复习题参考答案

复习题

1. 以计算机中所记录的形状参数与属性参数来表示图形的一种方法叫做_____，一般把它描述的图形叫做_____；而用具有灰度或颜色信息的点阵来表示图形的一种方法是_____，它强调图形由哪些点组成，并具有什么灰度或色彩，一般把它描述的图形叫做_____。

   A．参数法、图形、点阵法、图像　　B．点阵法、图像、参数法、图形

   C．参数法、图像、点阵法、图形　　D．点阵法、图形、参数法、图像

2. 下列设备中属于图形输出设备的是_____。

   ①鼠标②LCD③键盘④LED⑤打印机⑥扫描仪⑦绘图仪⑧触摸屏

   A．①③⑥⑧　　B．②④⑤⑦　　C．②⑤⑥⑦　　D．④⑥⑦⑧

3. 计算机显示器设备一般使用什么颜色模型_____。

   A．RGB　　B．CMYK　　C．HSV　　D．HLS

4. 灰度等级为 256，分辨率为 1024*1024 的显示器，至少需要的帧缓存容量为_____。

   A．512KB　　B．1MB　　C．2MB　　D．3MB

5. 多边形填充算法中，错误的描述是_____。

   A．有序边表算法对每个象素只访问一次，主要缺点是对各种表的维持和排序的耗费较大。

   B．边填充算法基本思想是对于每一条扫描线与多边形的交点，将其右方象素取补。

   C．边填充算法较适合于帧缓冲存储器的图形系统。

   D．边标志算法也不能解决象素被重复访问的缺点。

6. 在多边形的逐边裁剪法中，对于某条多边形的边（方向为从端点 S 到端点 P）与某条裁剪线（窗口的某一边）的比较结果共有以下四种情况，分别需输出一些顶点。请问哪种情况下输出的顶点是错误的_____。

   A．S 和 P 均在可见的一侧，则输出 S 和 P

   B．S 和 P 均在不可见的一侧，则输出 0 个顶点

   C．S 在可见一侧，P 在不可见一侧，则输出线段 SP 与裁剪线的交点

   D．S 在不可见的一侧，P 在可见的一侧，则输出线段 SP 与裁剪线的交点和 P

7. 下面关于反走样的论述哪个是错误的_____。

   A．提高分辨率　　B．把象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．增强图象的显示亮度

8. 按照所构造的图形对象来分，点、曲线、平面、曲面或实体属于_____，而山、水、云、烟等自然界丰富多彩的对象属于_____。

   A．规则对象、不规则对象　　B．规则对象、属性对象

   C．不规则对象、几何对象　　D．不规则对象、属性对象

9. 给定一系列型值点：$P_1P_2 \ldots P_{n-1}P_n$，怎样能够产生封闭的二次插值样条曲线_____。

   A．增加端点 $P_{n+1}=P_1$，$P_0=P_n$，$P_{n+2}=P_2$

   B．增加端点 $P_0=P_1$，$P_{n+1}=P_n$

C．将原端点替换为 $P_0'=2P_0-P_1$，$P_n'=2P_n-P_{n-1}$

D．将原端点替换为 $P_0'=P_0-2P_1$，$P_n'=P_n-2P_{n-1}$

10. 计算机图形学与计算几何之间的关系是_____。

A．学术上的同义词　　B．计算几何是计算机图形学的前身

C．计算机图形学以计算几何为理论基础　　D．两门毫不相干的学科

11. 关于触摸屏的叙述，正确的是_____。

A．触摸屏属于输入设备　　B．触摸屏属于输出设备

C．触摸屏属于输入/输出设备　　D．触摸屏不属于输入/输出设备

12. 种子填充算法中，正确的叙述是_____。

A．它是按扫描线的顺序进行象素点的填充

B．四连接算法可以填充八连接区域

C．四连接区域内的每一象素可以通过上下左右四个方向组合到达

D．八连接算法不能填充四连通区域

13. 多边形填充时，下述哪个论述是错误的_____。

A．多边形被两条扫描线分割成许多梯形，梯形底边在扫描线上，腰在多边形边上，相间排列

B．多边形与某扫描线相交得偶数个交点，交点间构成的线段分别在多边形内、外，相间排列

C．在判断点是否在多边形内时，一般通过在多边形外找一点，然后根据该线段与多边形的交　点数目为偶数即可认为在多边形内部，若为奇数则在多边形外部，而且不需考虑任何特殊情况

D．边的连贯性告诉我们，多边形的某条边与当前扫描线相交时，很可能与下一条扫描线相交

14. 下列有关 Bezier 曲线性质的叙述语句中，结论错误的是_____。

A．Bezier 曲线可用其特征多边形来定义。

B．Bezier 曲线不一定通过其特征多边形的各个顶点。

C．Bezier 曲线两端点处的切线方向必须与特征折线集（多边形）的相应两端线段走向一致。

D．n 次 Bezier 曲线，在端点处的 r 阶导数，只与 r 个相邻点有关。

15. 使用二维图形变换矩阵 $T=\begin{bmatrix}3 & 0 & 0\\ 0 & 1 & 0\\ 3 & 3 & 1\end{bmatrix}$，将产生变换的结果为_____。

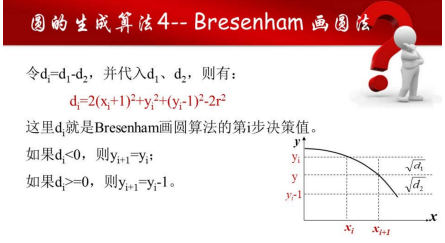
A．图形放大 3 倍

B．图形放大 3 倍，同时沿 X、Y 坐标轴方向各移动 3 个绘图单位

C．沿 X 坐标轴方向各移动 3 个绘图单位

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

16. 圆弧的 Bresenham 生成算法，通常把圆分成__8__个部分，如果用 d_1 和 d_2 来标识两个候选像素的 y 值与圆弧上理想 y 值的差值，则：$d_1=y_i^2-y^2=y_i^2-r^2+(x_i+1)^2$，$d_2=y^2-(y_i-1)^2=r^2-(x_i+1)^2-(y_i-1)^2$，令 $e_i=d_1-d_2$，当点(x_i, y_i)的 $e_i\geq 0$ 则下一点为 D，其坐标为



(x_i+1,__y_i- 1__)，若 $e_i<0$ 则下一点为 H，其坐标为(x_i+1,__y_i__)。

17. 有序边表扫描线算法中，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、__ __排序、__ __配对、__ __填色__。

18. 以下是中点画圆的一段程序，设半径 r＝100，颜色为 RED，试补充完成该程序。

```
#include <graphics.h>          /*图形函数库头文件声明*/
void circlePoint(int x,int y) {
      putpixel(200+x,200+y,RED);putpixel(200+y,200+x, RED) ;
      putpixel(200-y,200+x, RED);putpixel(200-x,200+y, RED);
    putpixel(200-x,200-y, RED) ; putpixel(200-y,200-x, RED);
      putpixel(200+y,200-x, RED);putpixel(200+x,200-y, RED);
}
void midCircle(int r) {
       int x,y,d;
       x=0;y=r;d=1-r;
       while(x<y) {
          circlePoint(x,y)        ;
            if(d<0) d+=2*x+3;
            else{ d+=2*(x-y)+5;y--;}
            x++;
       }
}
void main(){
       int gdriver,gmode;
       detectgraph(&gdriver,&gmode);
       initgraph(&gdriver,&gmode,"C:\\TURBOC2");
       MidBresenhamcircle(100);
       getch();
       closegraph();
 }
```

19. 在计算机图形学中，多边形有两种重要的表示方法：__顶点__表示和__点阵__表示。

20. 在区域编码裁剪算法中，如果线段 AB 的两个端点的编码__按位或为 0__则线段整体位于窗口内；如果两个端点的编码__按位与不为 0__则该线段整体位于窗口外。

21. 用一组型值点来指定曲线的形状时，形状完全通过给定的型值点列，用该方法得到的曲线称为曲线的__拟合__，而用控制点列来指定曲线的形状时，得到的曲线不一定通过控制点列，该方法称为曲线的__逼近__。

22. 由 5 个控制顶点 P_i(i=0,1,…4)所决定的 3 次 B 样条曲线，由__2__段 3 次 B 样条曲线段连接而成。

23. 齐次坐标系中，写出下列变换矩阵：整个图像放大 2 倍 $\begin{bmatrix}2&0&0\\0&2&0\\0&0&1\end{bmatrix}$；图像上移 10 个单

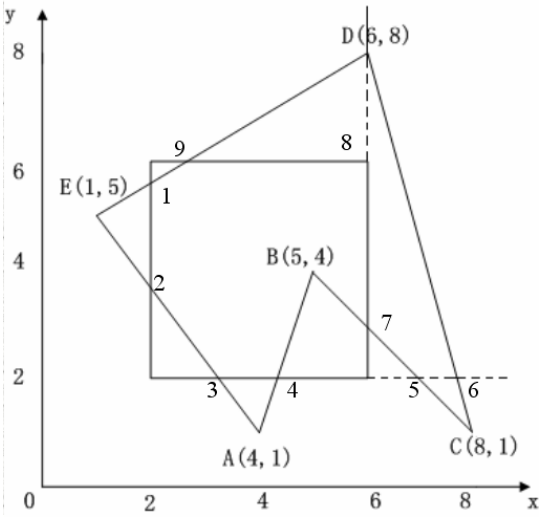
位和右移 5 个单位（y 轴垂直向上，x 轴水平向右）$\begin{bmatrix}1 & 0 & 5\\ 0 & 1 & 10\\ 0 & 0 & 1\end{bmatrix}$

24. 图形的输入设备有键盘、鼠标、触控笔、扫描仪（至少写三种）；图形的显示设备有CRT显示器、LCD、投影仪（至少写三种）。
25. 一个交互性的计算机图形系统应具有 计算 、存储、 、对话、 、输入、 、输出 等五方面的功能。
26. 齐次坐标表示就是用 n+1 维向量表示 n 维向量。
27. 与计算机图形学紧密相关的学科主要包括 图像处理、计算几何和计算机视觉模式识别，它们的共同点是以图形/图像在计算机中的表示方法为基础。（ T ）
28. 用离散量表示连续量时引起的失真现象称为走样。为了提高图形显示质量，减少或消除走样现象的技术称为反走样。（ T）
29. 彩色阴极射线管主要由红绿蓝三个彩色电子束亮度不同，进而组合成各种色彩。（F ）
30. DDA 直线算法的基本思想是：选定 x2－x1 和 y2－y1 中较大者作为步进方向，取该方向上的增量为一个象素单位，然后计算另一个方向的增量，其主要目的是考虑快速地生成直线。（ T ）
31. 边标志算法与活性边表算法比较，更适合于软件实现。（F ）
32. 四向种子填充算法可以用于填充八连通区域。（ F ）
33. 进行点的裁剪，已知窗口左下角坐标（50，100），右上角坐标（300，200），则点 P（150，300）在窗口内。（ F ）
34. 逐边多边形裁剪算法是一次完成对所有窗口边界的全部裁剪。（ F ）
35. DDA（微分方程法）是 Bresenham 算法的改进。（ F）
36. 直线的扫描转换，就是要找出显示平面上最佳逼近理想直线的那些象素的坐标值，并将这些象素置成所要求的颜色。 （T ）
37. 多边形裁剪实际就是直线段裁剪的简单组合。 （ F ）
38. 显式方程和参数曲线均可以表示封闭曲线或多值曲线。 （F ）
39. 参数法描述的图形叫图形；点阵法描述的图形叫图像。 （ T）
40. 0 阶参数连续性和 0 阶几何连续性的定义是相同的。 （T ）
41. Bezier 曲线可做局部调整。 （F）
42. 使用齐次坐标可以将 n 维空间的一个点向量唯一的映射到 n+1 维空间中。 （F）
43. 若要对某点进行比例、旋转变换，首先需要将坐标原点平移至该点，在新的坐标系下做比例或旋转变换，然后在将原点平移回去。 （T）
44. 在齐次坐标系中，若用矩阵来表示各种运算，则比例和旋转变换是矩阵乘法运算，而平移变换是矩阵加法运算。 （F ）
45. 用扫描线填充算法将顶点为 $P_1(2，2)$，$P_2(5，1)$，$P_3(11，3)$，$P_4(11，8)$，$P_5(5，5)$，$P_6(2，7)$的多边形填充，要求：

    （1）为每条扫描线建立一个新边表；

    （2）画出当扫描线 y=5 时的活性边表（AET 表）。
46. 如下图所示，裁减窗口为正方形，采用逐边裁件算法，依次按左、下、右、上的顺序，用四条窗口边界裁减多边形 ABCDE。试写出每条框口边界裁减后输出的新的多边形的

顶点序列。



47. 四边形 ABCD，顶点坐标分别是：A（4，1）、B（7，3）、C（7，7）和 D（1，4），求绕 P（5，4）点逆时针旋转 90 度的变换矩阵，并求出变换后的顶点坐标。

48. 已知控制顶点 $P=\{P_0 \quad P_1 \quad \ldots \quad P_3\}$ 可决定中间的一段抛物样条曲线 $p(t), t\in[0,0.5]$，证明抛物样条曲线能达到 C^1 连续。其中抛物样条曲线方程为：

$$P_i(t)=\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}\begin{bmatrix} -4 & 12 & -12 & 4 \\ 4 & -10 & 8 & -2 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} P_i \\ P_{i+1} \\ P_{i+2} \\ P_{i+3} \end{bmatrix} \quad t\in[0,0.5]$$

49. 考虑三个不同的光栅系统，分辨率依次为 640×480,1280×1024,2560×2048。欲存储每个像素 12 位，这些系统各需要多大的帧缓冲器（字节数）？

50. 已知矩形裁减窗口的四个顶点坐标分别是（150, 150），（150, 170），（180, 170），（180, 150），被裁减线段两个端点坐标分别为 C（160, 180），D（176, 140），试用编码裁减算法裁减此线段，写出计算过程。

45（仅供参考）

扫描线填充算法的基本思路是，先按顶点顺序连接多边形的边，并按照边的 y 坐标从小到大排序，建立一个边表（ET 表），然后从上到下扫描每条扫描线，对于每条扫描线，将和它相交的边按照交点的 x 坐标从小到大排序，建立一个活性边表（AET 表），然后依次取出相邻两个边的交点，填充两个交点之间的像素。

下面是按照这个思路填充顶点为 P1(2，2)，P2(5，1)，P3(11，3)，P4(11，8)，P5(5，5)，P6(2，7)的多边形的过程：

1. 建立边表（ET 表）：

```markdown
ymin  ymax  x   dx   dy
-----------------------
 1     2    5  3/4  -1
 2     2    2  -3/5 -1/3
 2     3   11     0  1
 3     5   5  -2/3  2/3
 5     7   2  3/5   1/3
 5     8   11     0  1
 8     8    -inf inf  -inf
```

2. 从上到下扫描每条扫描线，建立活性边表（AET 表）：

扫描线 y=5 时，与之相交的边是 P2P5 和 P4P5，因此活性边表为：

```markdown
x      dx   dy  1/m
------------------
 5     -2/3 2/3 -3/2
11      0   1    inf
```

其中，1/m 表示斜率的倒数。

3. 依次取出相邻两个边的交点，填充两个交点之间的像素：

```markdown
交点1: x = 3, P2P5 和 P4P5 的交点，对应 AET 表中的第一条边，将其加入活性边表，AET 表变为:

x      dx   dy  1/m
------------------
 5     -2/3 2/3 -3/2
11      0   1    inf
 3     -3/5 1/3 -3/5

交点2: x = 5, P2P5 的终点和 P3P4 的起点的交点，对应 AET 表中的前两条边，将它们从活性边表中删除，AET 表变为:

x      dx   dy  1/m
------------------
11      0   1    inf
 3     -3/5 1/3 -3/5

填充像素 (3, 5) 和 (4, 5)，颜
```

色取决于多边形内部的填充规则。如果采用奇偶填充规则，那么由于 (3, 5) 和 (4, 5) 都被覆盖了两次，所以需要交替填充两种颜色，比如白色和黑色。如果采用非零环绕数填充规则，那么可以从多边形内部的任意一点出发，沿着任意方向绘制一条射线，如果这条射线与多边形的边相交的次数是奇数，则说明这个点在多边形内部，需要填充。否则这个点在多边形外部，不需要填充。

由于题目只要求我们画出当扫描线 y=5 时的活性边表（AET 表），因此我们只需要在第二步中找到与扫描线相交的边，并将它们加入活性边表即可，不需要进行后续的像素填充。

下面是扫描线 y=5 时的活性边表（AET 表）：

```markdown
x      dx   dy  1/m
------------------
 5     -2/3 2/3 -3/2
11      0   1    inf
```

其中，dx 和 dy 分别表示边的斜率，1/m 表示斜率的倒数。

由于扫描线 y=5 与边 AB 和 CD 相交，因此它们需要加入活性边表。计算出它们的斜率和斜率的倒数：

```scss
边 AB: dx = (5 - 2) / (2 - 2) = -3, dy = (1 - 2) / (2 - 2) = -1, 1/m = -1/(-3
边 CD: dx = (11 - 5) / (3 - 2) = 6, dy = (8 - 5) / (3 - 2) = 3, 1/m = 3/6 = 1
```

将它们加入活性边表，并按照 x 坐标排序：

```markdown
x      dx   dy  1/m
------------------
 5     -3   -1    1/3
11      6    3    1/2
```

注意，斜率为正的边在排序时需要放在斜率为负的边的前面。这是因为当两条边在同一水平线上相交时，斜率为正的边会先进入多边形内部，而斜率为负的边则会先退出多边形。因此，我们需要先处理斜率为正的边，再处理斜率为负的边。

这就是扫描线 y=5 时的活性边表（AET 表）了。

47（仅供参考）

要求一个点绕另一个点逆时针旋转90度的变换矩阵，可以按照以下步骤进行：

1. 将原始点P以旋转中心点Q为原点平移至坐标系的原点。
2. 将平移后的点P绕原点逆时针旋转90度，得到新点P'的坐标。
3. 将新点P'以旋转中心点Q为原点反向平移回原来的位置，即可得到变换矩阵。

根据这个步骤，可以得到四边形绕P（5，4）点逆时针旋转90度的变换矩阵：

1. 平移点P（5，4）至坐标系原点：P' = (0, 0) - (5, 4) = (-5, -4)
2. 将P'绕原点逆时针旋转90度得到P''，由于是逆时针旋转，所以x和y的值交换并取相反数：P'' = (-P'.y, P'.x) = (4, -5)
3. 将P''以点P（5，4）为原点反向平移回原来的位置：P''' = P'' + (5, 4) = (9, -1)

因此，绕P（5，4）点逆时针旋转90度的变换矩阵为：

arduino Copy code

```
cosθ  -sinθ   x
sinθ   cosθ   y
  0      0    1
```

其中，θ为旋转角度，x和y为平移距离，代入上面的计算结果可以得到变换矩阵：

Copy code

```
0  -1  9
1   0  -1
0   0  1
```

接下来，将四边形的顶点坐标带入变换矩阵，即可得到变换后的顶点坐标：

scss Copy code

```
A' = (4, 1, 1) * M = (-3, 13, 1)
B' = (7, 3, 1) * M = (-1, 10, 1)
C' = (7, 7, 1) * M = (3, 10, 1)
D' = (1, 4, 1) * M = (-2, 12, 1)
```

因此，四边形绕P（5，4）点逆时针旋转90度后的顶点坐标分别为：A'(-3, 13)，B'(-1, 10)，C'(3, 10)和D'(-2, 12)。

49 解：640×480需要的帧缓存为$640\times480\times12/8=450KB$

1280×1024需要的帧缓存为$1280\times1024\times12/8=1920KB$

2560×2048需要的帧缓存为$2560\times2048\times12/8=7680KB$

50（仅供参考）

编码裁剪算法是一种常见的直线裁剪算法，可用于确定给定线段是否被矩形裁剪窗口所裁剪，并计算出裁剪后的线段。

以下是裁剪算法的步骤：

1. 将矩形裁减窗口的四个顶点分别编码为四位二进制数，其中左边的位表示该点是否在窗口左边界的左侧，右边的位表示该点是否在窗口右边界的右侧，上方的位表示该点是否在窗口上边界的上方，下方的位表示该点是否在窗口下边界的下方。例如，点（150, 150）编码为 1001，因为它在窗口左侧和下方。
2. 将线段的两个端点 C 和 D 分别编码为二进制数。
3. 检查端点 C 和 D 是否都在矩形窗口的一个侧面之外，如果是，则线段完全在窗口之外，不需要裁剪，算法结束。
4. 如果 C 和 D 都在矩形窗口内部，则线段完全在窗口内部，不需要裁剪，算法结束。
5. 否则，需要裁剪线段。将 C 和 D 的编码进行逻辑运算，得到一个新的编码，表示 C 和 D 都在窗口某一侧的情况，具体为将 C 和 D 的编码做按位与运算，得到的结果如果不为 0000，则表示 C 和 D 都在某个侧面之外。
6. 如果新编码为 0000，则线段与窗口某侧的交点就是 C 和 D 本身。此时，线段被裁剪为原线段。
7. 否则，找到新编码为非零值的那一侧，计算线段与该侧的交点，并将该点作为 C 或 D 的新值，重复步骤 5 到步骤 7，直到线段被裁剪为止。

根据上述步骤，将给定的矩形窗口和线段 C（160, 180），D（176, 140）代入算法中，可以得到如下计算过程：

Regenerate response

1. 将矩形窗口的四个顶点编码为二进制数：

顶点 A（150, 150）：1001
顶点 B（150, 170）：1000
顶点 C（180, 170）：0000
顶点 D（180, 150）：0010

2. 将线段的两个端点编码为二进制数：

端点 C（160, 180）：0110
端点 D（176, 140）：0001

3. 检查端点 C 和 D 是否都在

矩形窗口的一个侧面之外。根据编码，发现 C 在矩形窗口左侧和上方，而 D 在矩形窗口右侧和下方，因此 C 和 D 都在窗口某一侧的情况。

4. 将 C 和 D 的编码进行逻辑运算，得到一个新的编码 0010。
5. 新编码不为 0000，因此找到新编码为非零值的那一侧，即 D 所在的侧面。由于 D 在窗口右侧和下方，因此选择窗口右侧进行裁剪。
6. 计算线段 CD 与窗口右侧的交点。由于窗口右侧的 x 坐标为 180，因此将 x 坐标设为 180，根据线段的斜率可以计算出交点的 y 坐标为 160。
7. 将交点（180, 160）作为 D 的新值，将其编码为二进制数 0010，重复步骤 3 到步骤 5，直到线段被裁剪为止。
8. 由于 D 的新值（180, 160）已经在窗口内部了，因此此时线段被裁剪为 C（160, 170），D（180, 160）。

综上所述，根据编码裁剪算法的步骤，可以对给定的线段进行裁剪，得到裁剪后的线段 C（160, 170），D（180, 160）。