

Beantworten Sie die Fragen in Aufgabe 1 mit einer kurzen, prägnanten Antwort. In der Regel reichen ein bis zwei Wörter.

### Aufgabe 1 (15 Punkte)

1. Welcher Punkt im  $\mathbb{R}^3$  wird durch die homogenen Koordinaten  $(5, 4, 3, 2)$  beschrieben?

2. Mit welchem Verfahren lässt sich beim Bresenham-Algorithmus für Linien vermeiden, dass Linien mit unterschiedlicher Steigung unterschiedlich dick erscheinen?

3. In welchem Koordinatensystem wird der z-Buffer-Algorithmus durchgeführt?

4. Welche Farben sprüht ein Tintenstrahldrucker auf das Papier, wenn Sie rot drucken möchten?

5. Welche Dimension hat die Dithermatrix  $D_4$ ?

6. Bei welchem Shading-Verfahren wird der Mach-Band-Effekt weitestgehend vermieden?

7. Welcher Bereichscode nach Cohen & Sutherland gibt an, dass man sich unten rechts vom Clipping-Fenster befindet?

8. Welche Bedingung muss gelten, damit ein *Flat Face* für den Betrachter sichtbar ist?

9. In welchem Farbraum findet die JPEG-Komprimierung statt?

10. Wie ist die *View Plane Distance* definiert?

11. Bei welcher Reflexion entsteht ein Glanzpunkt?

12. Wie lauten die homogenen Koordinaten des Punktes  $(x, y, z, 1)$  nach orthogonaler Parallelprojektion?

13. Welche Form hat das *Frustum* vor Transformation ins NPC?

14. Wieviele verschiedene Bernsteinpolynome ergeben sich, wenn man eine quadratische Bezierkurve definieren möchte?

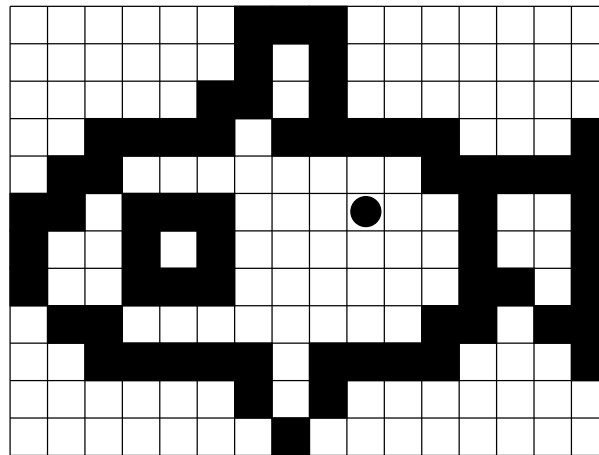
15. Wodurch unterscheidet sich *Displacement Mapping* von anderen Texturing-Verfahren?

**Aufgabe 2 (3 Punkte)**

Wie viele Pixel setzt der Bresenham-Algorithmus beim Zeichnen eines Kreises mit dem Radius  $r$ ?

**Aufgabe 3 (3 Punkte)**

Füllen Sie das Gebiet mit dem 8-Way-Stepping beginnend bei dem Punkt ●:

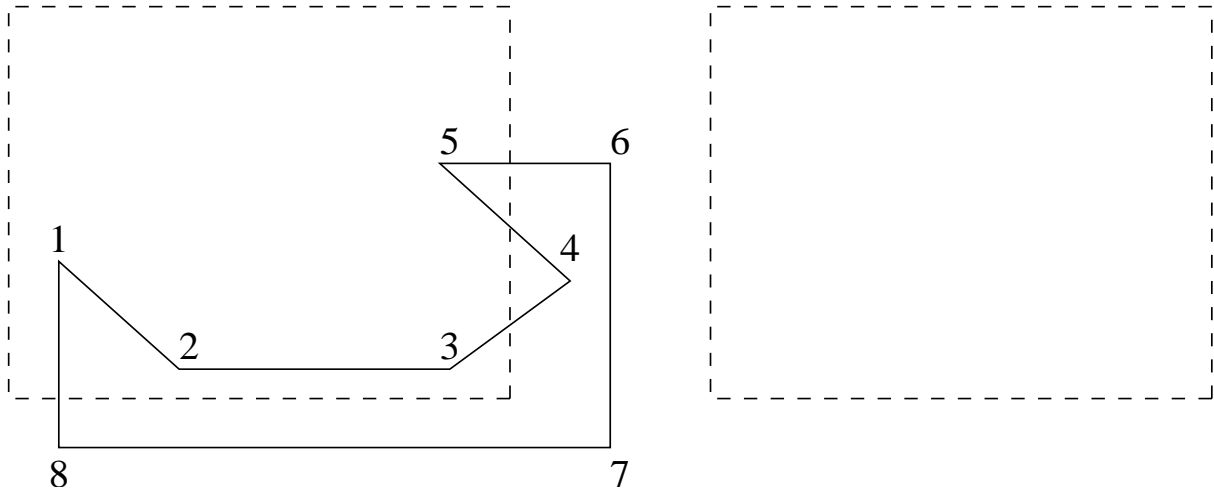
**Aufgabe 4 (2 Punkte)**

Geben Sie eine geeignete  $4 \times 4$ -Dithermatrix für ordered Dithering an:

$$\begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

**Aufgabe 5 (3 Punkte)**

Clippen Sie das Polygon nach dem Algorithmus von Sutherland & Hodgman an dem gestrichelten Rechteck und nummerieren Sie die Eckpunkte des neuen Polygons durch.

**Aufgabe 6 (2 Punkte)**

Konstruieren Sie zu den gegebenen 4 Punkten nach dem Algorithmus von deCasteljau den Kurvenpunkt der kubischen Bézier-Kurve zum Wert  $t = \frac{3}{4}$ .

1 ●

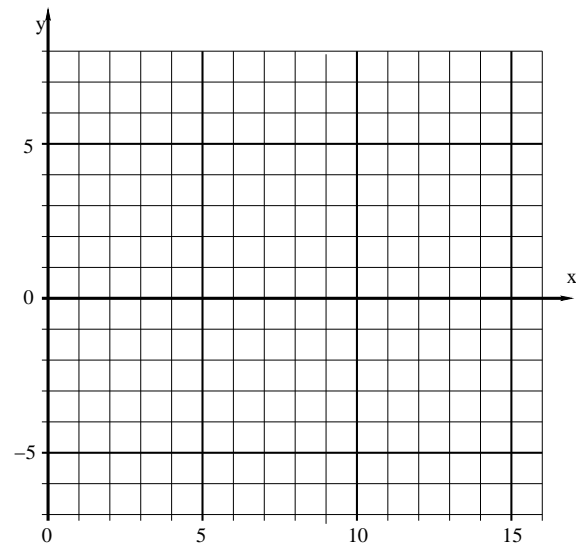
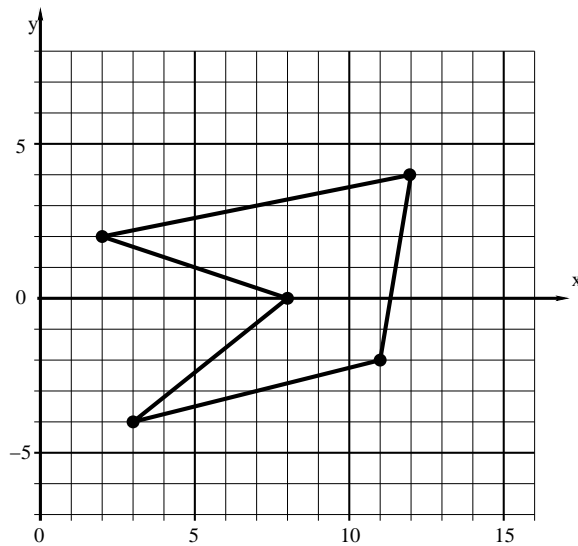
4 ●

2 ●

3 ●

**Aufgabe 7 (8 Punkte)**

Scheren Sie das Polygon in der folgenden Zeichnung um den Scherwert  $Sch_x = 0,5$  und spiegeln Sie es anschließend an der X-Achse. Tragen Sie das Ergebnis in das rechte Koordinatensystem ein. (5 Punkte)

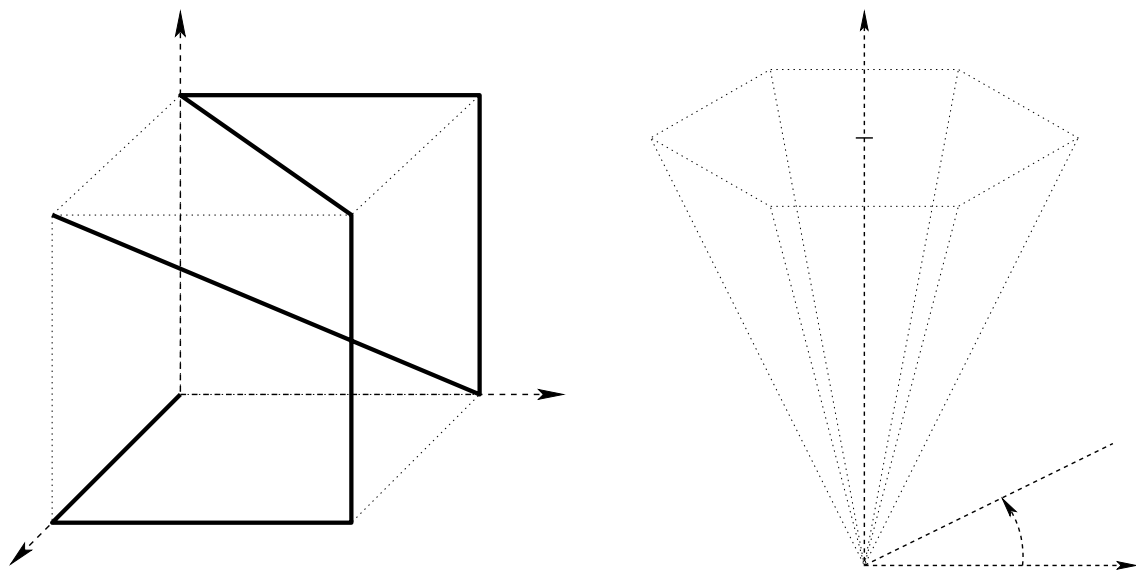


Wie lautet die Transformationsmatrix der Gesamttransformation? (3 Punkte)

**Aufgabe 8 (6 Punkte)**

Gegeben ist ein RGB-Würfel und eine HSV-Pyramide.

1. Beschriften sie die HSV-Pyramide und den RGB-Würfel mit den sechs Farben (R, G, B, C, M, Y) sowie schwarz und weiß (S, W). (2 Punkte)
2. Im RGB-Würfel finden Sie einen Linienzug, der einmal durch jede der Farben rot, grün, blau, cyan, magenta und gelb, sowie schwarz und weiß verläuft. Zeichnen Sie den Weg durch diese Farben mit einem Lineal in die HSV-Pyramide ein. (4 Punkte)

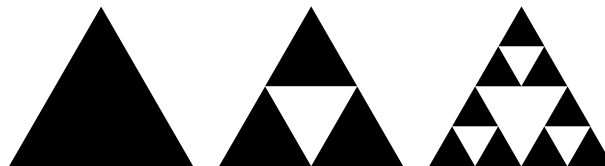
**Aufgabe 9 (3 Punkte)**

Geben sie die Farbe **Cyan** als Tripel in dem jeweiligen Farb-Modell an:

RGB	([0 ... 100%], [0 ... 100%], [0 ... 100%])	( , , )
CMY	([0 ... 100%], [0 ... 100%], [0 ... 100%])	( , , )
HSV	([0 ... 360°], [0 ... 100%], [0 ... 100%])	( , , )

**Aufgabe 10 (2 Punkte)**

Das *Sierpinski-Dreieck* ist ein Fraktal, das dadurch entsteht, dass man wiederholt ein Vorgängerdreieck in vier gleichgroße Dreiecke zerlegt und das mittlere davon entfernt (siehe Abbildung).



Zwei Iterationsschritte des Sierpinski-Dreiecks

Welche Fraktale Dimension hat das *Sierpinski-Dreieck*?

**Aufgabe 11 (4 Punkte)**

Geben Sie die homogene Matrix für die Rotation im Uhrzeigersinn um die  $x$ -Achse in einem dreidimensionalen rechtshändigen Koordinatensystem um den Winkel  $\alpha$  an.

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} | & | & | & | \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$$

**Aufgabe 12 (12 Punkte)**

Ein Objekt soll an einer Geraden mit der Gleichung  $f(x) = \frac{3}{4} \cdot x - 1$  im  $\mathbb{R}^2$  gespiegelt werden. Lösen Sie dieses Problem durch eine Sequenz von Transformationen und schreiben Sie die benötigten Teilmatrizen mit **konkreten numerischen Werten** auf und geben Sie die Reihenfolge an, in der sie miteinander multipliziert werden müssen.



**Aufgabe 13 (9 Punkte)**

Im WC sind gegeben:

$$VRP = (4, 3, 2)$$

$$PRP = (4, -2, 2)$$

$$VUP = (6, 3, 2)$$

Wie lautet die Transformationsmatrix für einen Punkt in homogenen Koordinaten  $(x, y, z, w)$  vom **VRC ins WC**? (7 Punkte)


Mit welcher homogenen Transformationsmatrix werden die Normalenvektoren vom **WC ins VRC** abgebildet? (2 Punkte)


**Aufgabe 14 (6 Punkte)**

Die Viewing-Pipeline zur Projektion von dreidimensionalen Szenen auf ein zweidimensionales Ausgabegerät ist in mehrere Schritte unterteilt. Welche Namen haben diese Schritte und was passiert im jeweiligen Schritt?

Name des Schrittes	Aktion im Schritt



**Aufgabe 16 (6 Punkte)**

Ein Face der Größe 7 x 7 Pixel soll trianguliert und mit Gouraud-Shading in Graustufen gerendert werden. Die Graustufen sind ganzzahlige Zahlen im Intervall  $[0; 255]$ . Tragen Sie in das untenstehende Face die interpolierten Graustufenwerte an den entsprechenden Stellen ein. Die aus der Beleuchtung resultierenden Grauwerte in den Eckpunkten des Faces sind bereits eingetragen.

42						18
6						66

**Aufgabe 17 (10 Punkte)**

Auf den folgenden Seiten finden Sie den Quelltext eines OpenGL-Programms, das bereits eine fertige statische Szene definiert, die jedoch keine Objekte enthält.

Vervollständigen Sie die `display`-Funktion, so dass in der Szene ein 'Y' aus drei Quadern dargestellt wird (siehe Abbildung). Der untere Quader hat die Höhe 5, die beiden anderen Quader die Länge 4. Die beiden oberen Quader sind jeweils um  $45^\circ$  nach rechts und links geneigt. Das 'Y' soll auf der XZ-Ebene stehen, d.h. der untere Quader beginnt im Ursprung.

Verwenden Sie **ausschließlich** die folgenden Funktionsaufrufe **mit evtl. angepassten Parametern**:

```
glutWireCube(1.0);           // Drahtgitter eines Quadrates mit
                              // Kantenlaenge 1 und Zentrum in (0,0,0)

glTranslatef(0.0, 2.0, 0.0); // Translation um (0,2,0)
glScalef(1.0, 2.0, 1.0);    // Skalierung um Faktor (1,2,1)
glRotatef(90, 0.0, 0.0, 1.0); // Rotation um 90 Grad um die z-Achse,
                              // definiert durch (0,0,1)

glPushMatrix();
glPopMatrix();
```

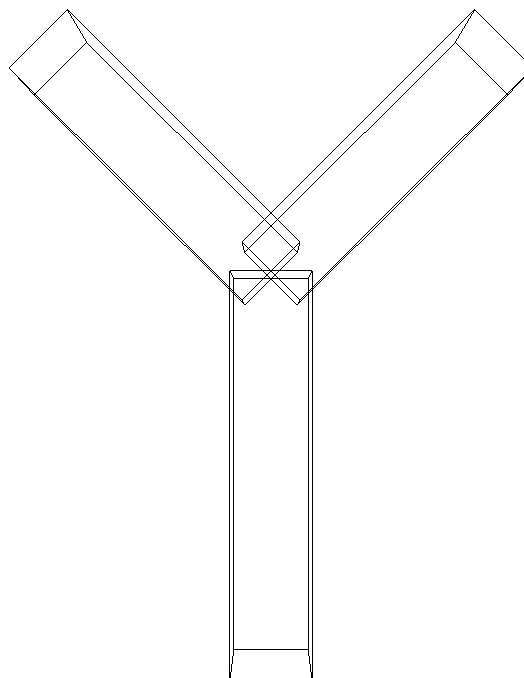


Abbildung 1: Die zu erstellende Szene

---

```
#include <GL/glut.h>           /* OpenGL Utility Toolkit */
#include <stdlib.h>             /* C-library */

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT); /* reset Pixels */
    glColor3f(0.0, 0.0, 0.0); /* Farbe weiss */
    gluLookAt(0.0, 4.0, 10.0, /* PRP */
              0.0, 4.0, 0.0, /* VRP */
              0.0, 1.0, 0.0); /* VUP */

    glFlush(); /* Puffer anzeigen */
}
```

```
void init(void)
{
    glClearColor(1.0, 1.0, 1.0, 0.0);      /* blauer Hintergrund      */
    glShadeModel(GL_FLAT);                 /* Flatshading            */
    glMatrixMode(GL_PROJECTION);           /* ab jetzt: Projektion   */
    glLoadIdentity();                     /* lade Einheitsmatrix    */
    gluPerspective(60.0, 1.33, 1.0, 50.0); /* Blickwinkel, w/h, near, far */
    glMatrixMode(GL_MODELVIEW);           /* ab jetzt: Modelview    */
}

int main (int argc, char ** argv)         /* Hauptprogramm          */
{
    glutInit(&argc, argv);                 /* initialisiere GLUT     */
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB); /* double buffer, true color */
    glutInitWindowSize(800, 600);         /* initiale Fenstergrösse */
    glutInitWindowPosition(0,0);          /* initiale Fensterposition */
    glutCreateWindow("Y");                 /* Fenster mit Aufschrift */
    init();                                 /* rufe init auf          */
    glutDisplayFunc(display);              /* registriere display    */
    glutMainLoop();                        /* beginne Event-Schleife */
    return 0;
}
```