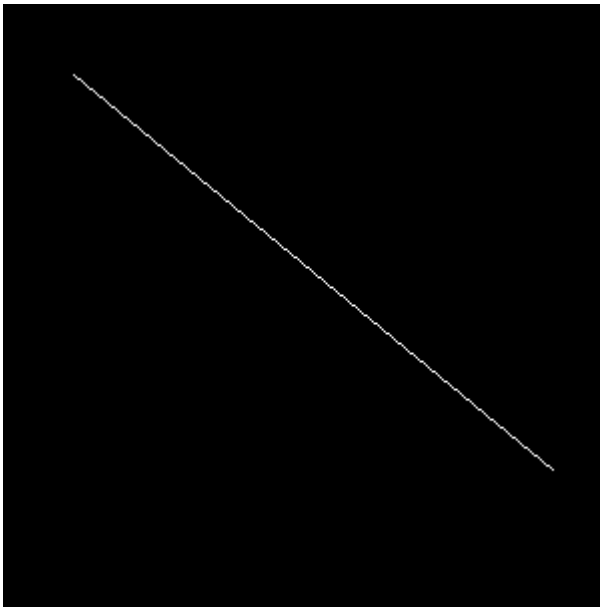
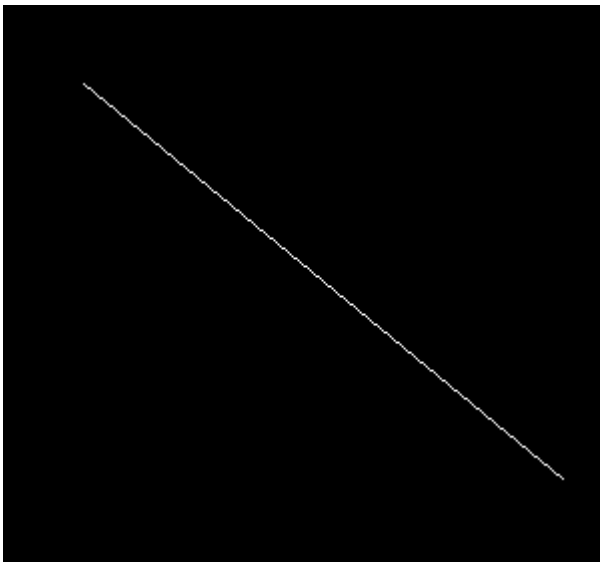

```
GLubyte cLetter[24] = {
    0xff, 0xc0, 0xff, 0xc0, 0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00,
    0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00,
    0xff, 0xc0, 0xff, 0xc0};

GLubyte hLetter[24] = {
    0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0,
    0xff, 0xc0, 0xff, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0,
    0xc0, 0xc0, 0xc0, 0xc0};

GLubyte aLetter[24] = {
    0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0,
    0xff, 0xc0, 0xff, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0, 0xc0,
    0xff, 0xc0, 0xff, 0xc0};
```

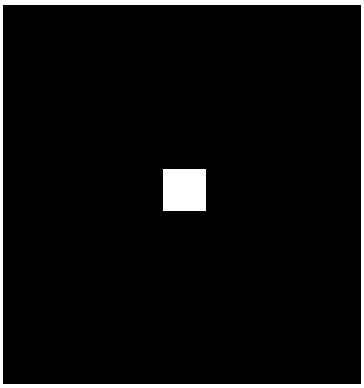
```
void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f (1.0, 0, 1.0);
    glRasterPos2i (200, 200);
    glBitmap (10, 12, 0.0, 0.0, 11.0, 0.0, cLetter);
    //On ne peut pas colorer plusieurs lettres d'une couleur différente
    glColor3f (1.0, 1, 0);
    glBitmap (10, 12, 0.0, 0.0, 11.0, 0.0, hLetter);
    glBitmap (10, 12, 0.0, 0.0, 11.0, 0.0, aLetter);
    glFlush();
}
```



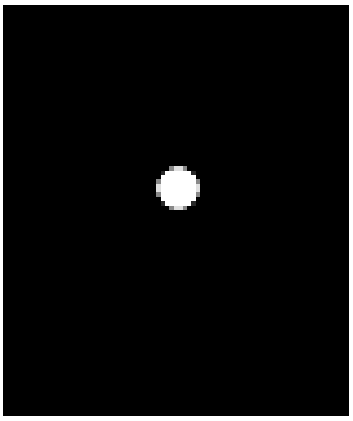
```
void displayLines(){
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);
    glBegin(GL_LINE_STRIP);
    glVertex3f(124,348,0);
    glVertex3f(364,150,0);
    glEnd();
    glFlush();
}
```



```
void displayLines(){
    //Aucunce différence sous Linux car les fonctions ne sont pas implémentées
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);
    glEnable(GL_LINE_SMOOTH);
    glEnable(GL_BLEND);
    glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
    glHint(GL_LINE_SMOOTH_HINT, GL_DONT_CARE);
    glBegin(GL_LINES);
    glVertex3f(124,348,0);
    glVertex3f(364,150,0);
    glEnd();
    glFlush();
}
```



```
void displayPoints(){
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);
    glPointSize(10);
    glBegin(GL_POINTS);
    glVertex3f(124,348,0);
    glEnd();
    glFlush();

}
```



```
void displayPoints(){
    //Maitenant on voit la différence
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);
    glPointSize(10);
    glEnable(GL_POINT_SMOOTH);
    glEnable(GL_BLEND);
    glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
    glHint(GL_POINT_SMOOTH_HINT, GL_DONT_CARE);
    glBegin(GL_POINTS);
    glVertex3f(124,348,0);
    glEnd();
    glFlush();

}
```

Pour l’algo Cohen voir code source.