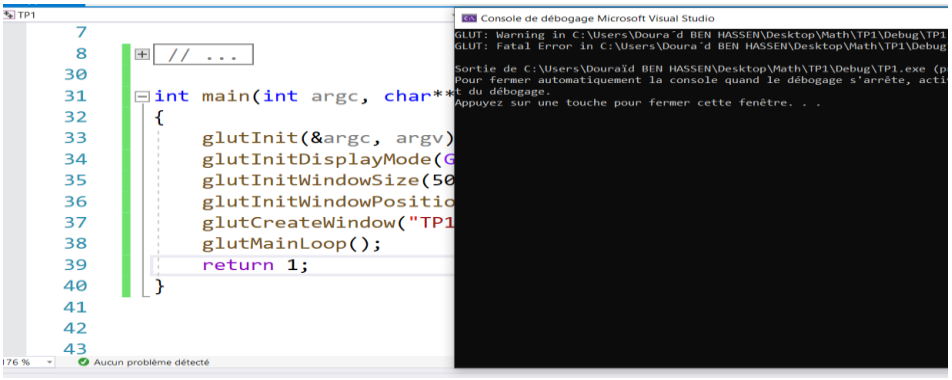
BEN HASSEN
Douraïd

Groupe 2

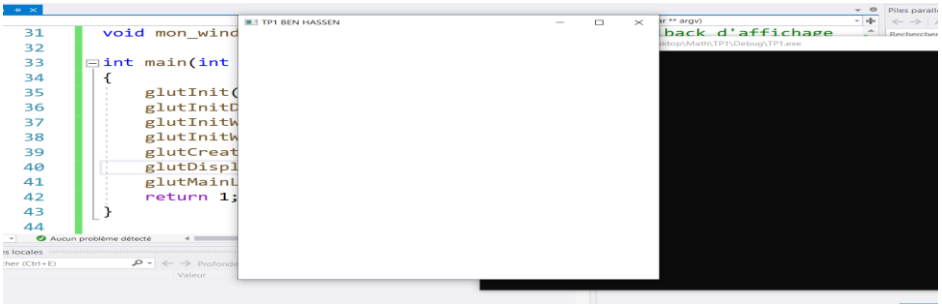
Exercice 1 :

a) Rien ne s'affiche, il faut une fonction d'affichage

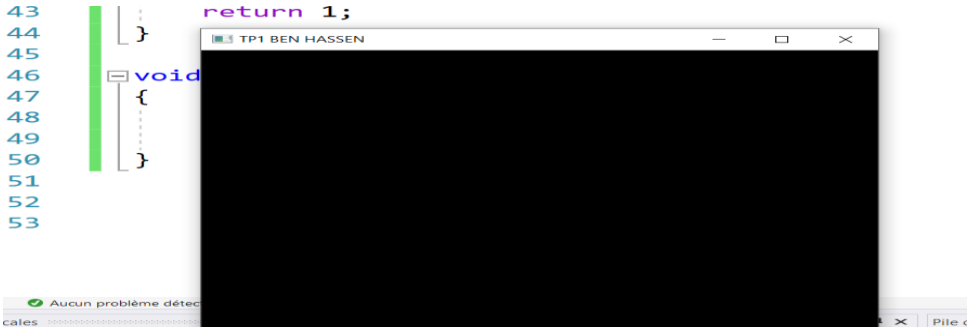


Exercice 2 :

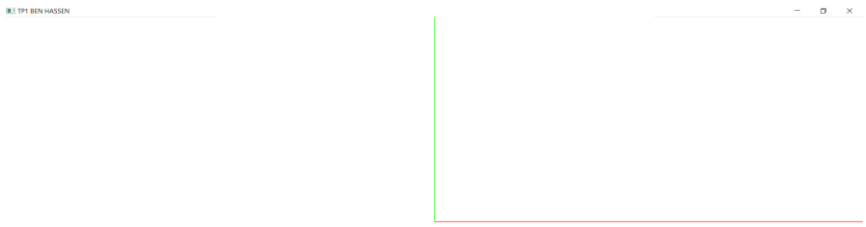
a) Une fenêtre blanche s'affiche



b) Fenêtre noire



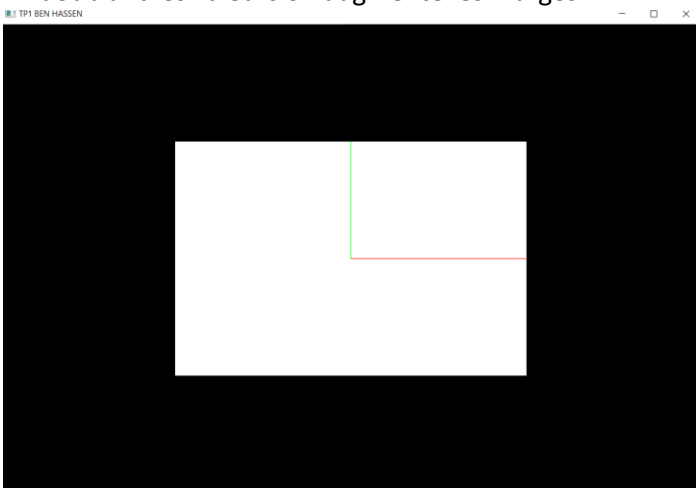
c) Fond blanc avec axes x y au centre



Exercice 3 :

a) Projection sans marge :

b) En doublant les valeurs on augmente les marges :



Exercice 4 :

a) Les couleurs ne changent pas

b) Les couleurs changent

c) Les couleurs changent et s’actualisent

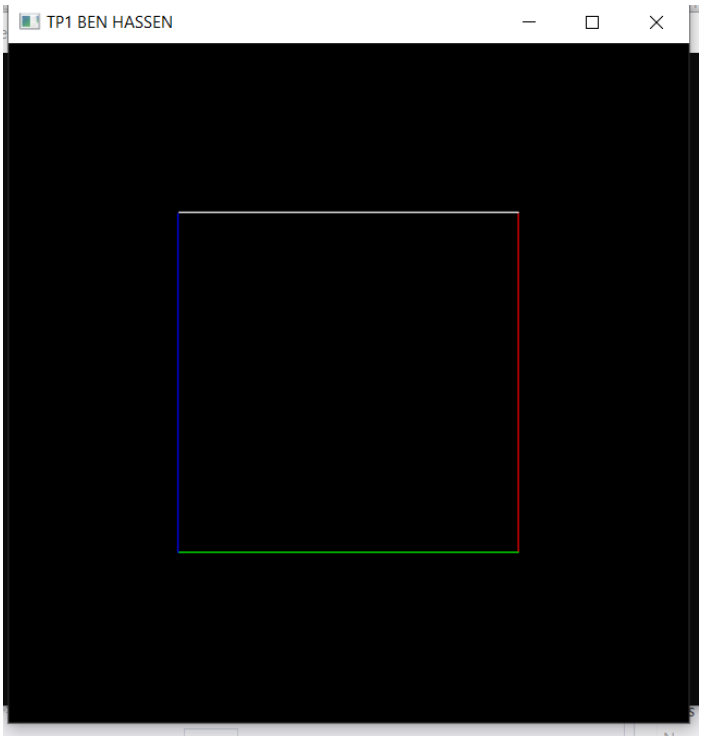
Exercice 5 :

```
int color = 0; // 0:noir 1:rouge 2:vert 3:bleu 4:blanc
```

```
void mon_mouse(int button, int state, int x, int y) {
    if (state == 0) {
        switch (color)
        {
        case 0:
            if (button == 0) {
                color = 1;
                glClearColor(1, 0, 0, 1);
                glClear(GL_COLOR_BUFFER_BIT);
                glFlush();
                glutPostRedisplay();
            }
            else {
                color = 4;
                glClearColor(1, 1, 1, 1);
                glClear(GL_COLOR_BUFFER_BIT);
                glFlush();
                glutPostRedisplay();
            }
            break;
```

Exercice 6 :

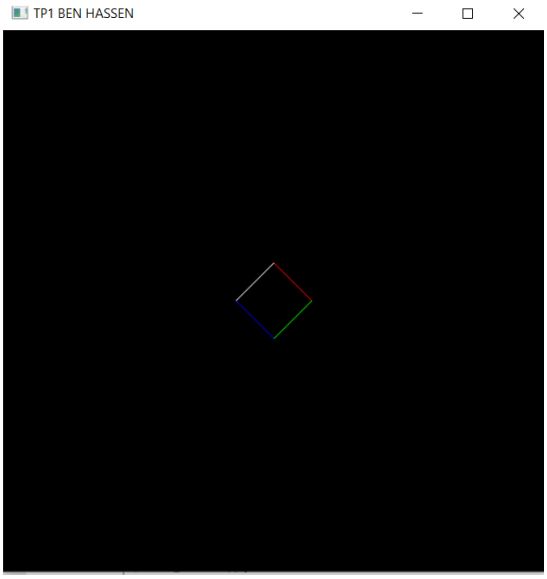
a)



4 sommets colorés, 1 blanc, 2 rouge, 3 vert et 4 bleu

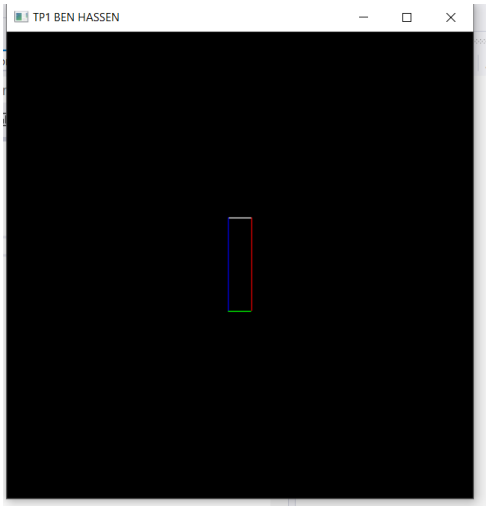
Cas 1 :

glRotatef(0.45, 0, 0, 1) : rotation des sommets de 45°



glTranslatef(-5, 5, 0) décalage de -5 en x et +5 en y

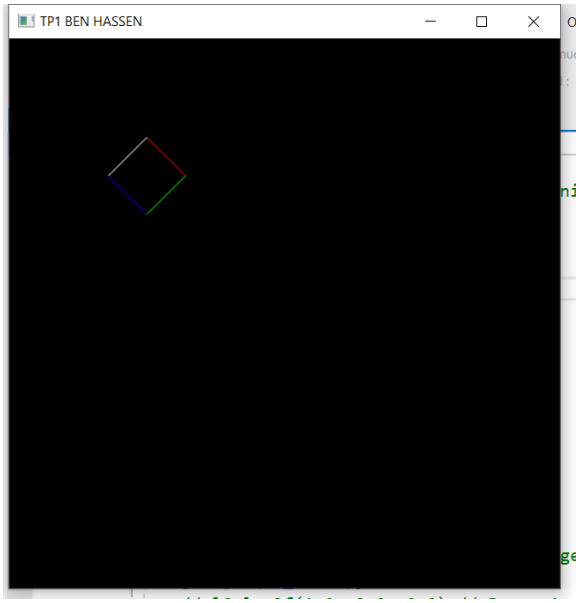
glScalef(0.5, 2) : la largeur divisé 2 et la hauteur est multiplié par 2



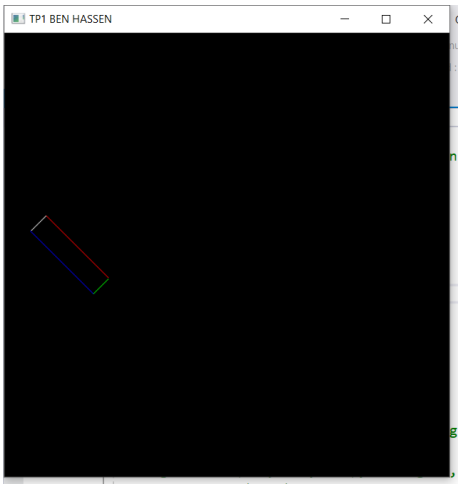
Cas 2 :

glRotatef() + glTranslatef() : la rotation de 45° se fait avant la translation en -5, 5

glTranslatef() + glRotatef() : la translation en -5, 5 se fait avant la rotation de 45°

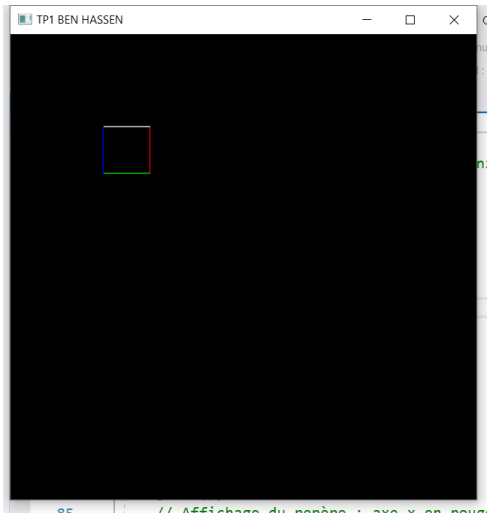


glRotatef() + glTranslatef() + glScalef() : je fais la rotation de 45° en 1er, ensuite la translation de -5,5 et enfin le redimenssionement de 0.5,2

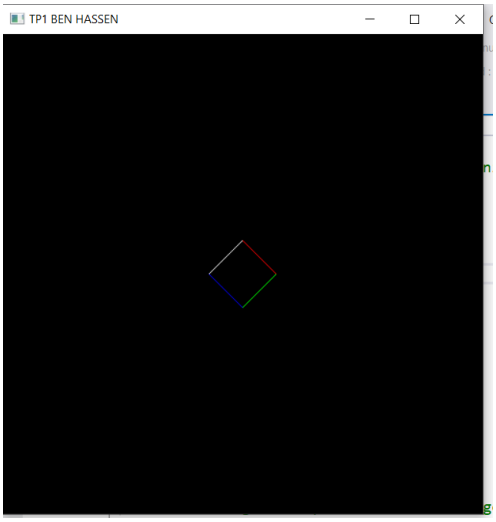


Cas 3 :

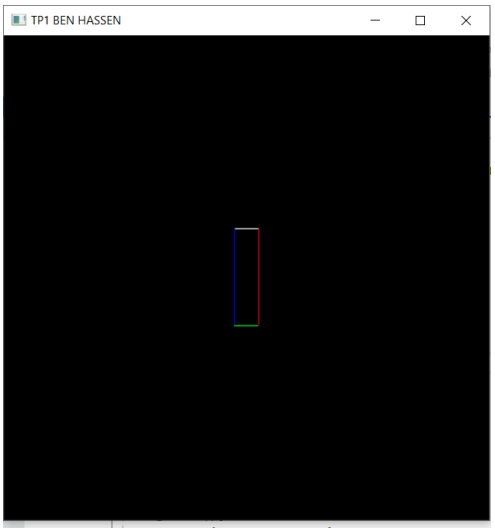
glRotatef() + glTranslatef() la translation en -5, 5 est faite avant la rotation de 45°



glTranslatef() + glRotatef() la rotation de 45° est faite avant la translation en -5, 5



glRotatef() + glTranslatef() + glScalef() la rotation de 45°, ensuite la translation en -5,5 suivie du redimenssionement en 0.5, 2

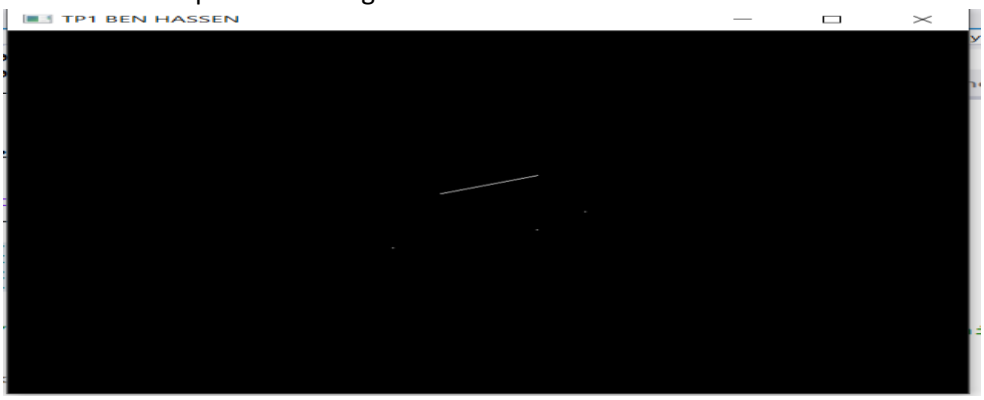


Avec `glLoadIdentity(), la prochaine transformation se fait depuis la matrice d'origine`

Exercice 7 :

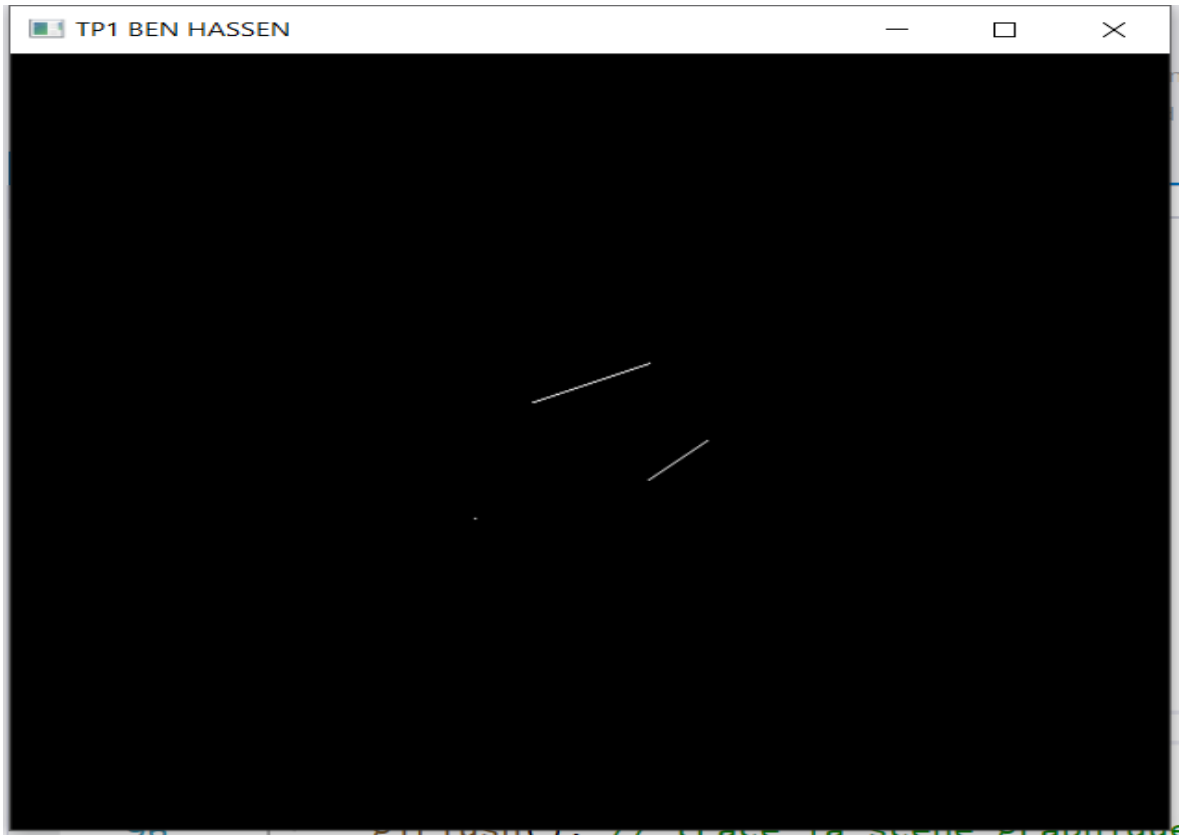
a)

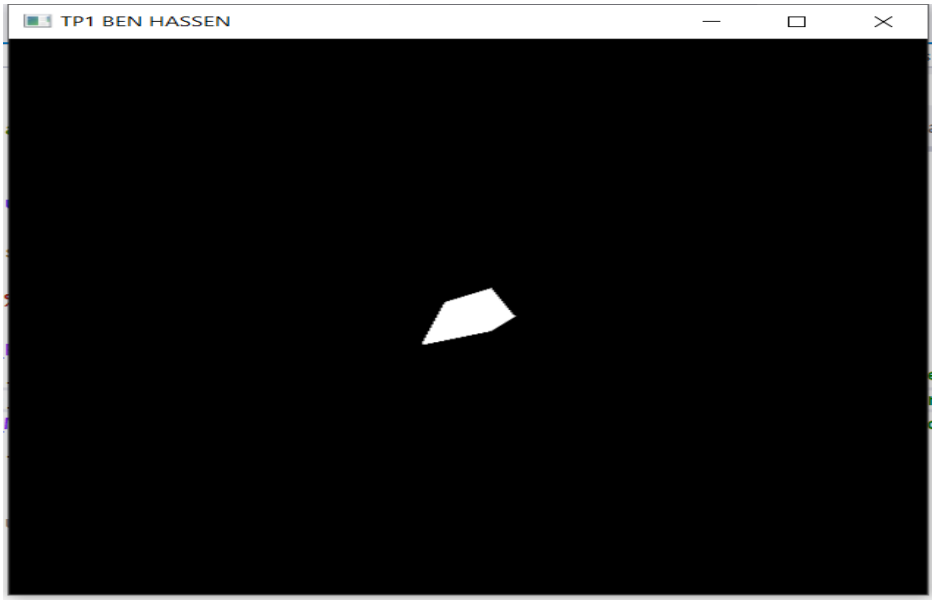
Cas 1 : On a les points et le segment S0S1



Cas 2 : On a les points et le segment S0S1 + le segment S2S3

Cas 3 : Il reste qu'un seul point

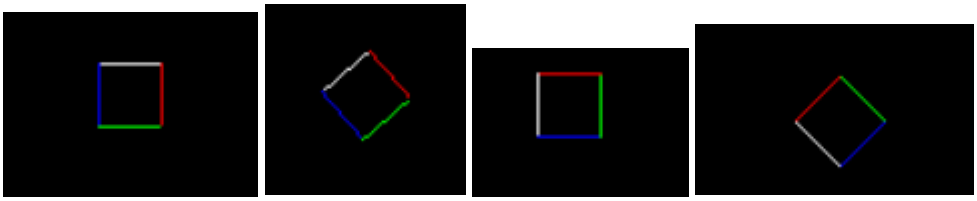




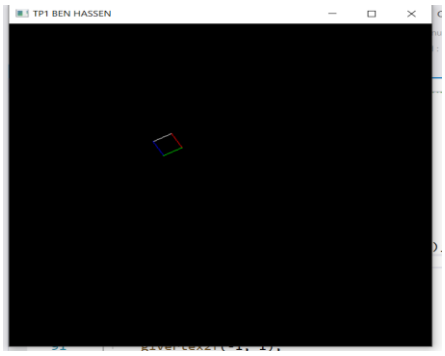
Avec translation : j'ai fait en sorte d'avoir une translation (2 en 2) sur chaque dessin pour avec toutes les étapes (segments et remplissage)

Exercice 8 :

Cas 1 : je fais tourner le carré sur lui-même



Cas 2 : je fais une translation en -5,5, puis je fais tourner le carré sur lui-même



Cas 3 : je fais une rotation sur un angle alpha puis une translation en -5,5, le carré tourne en rond et sur lui même

Exercice 9 :

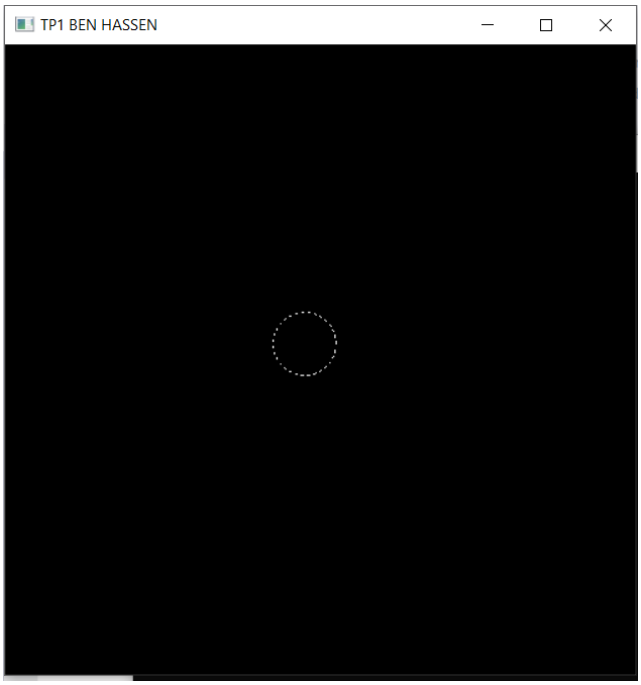
a : 0 → 2π

```
X = posX + (rayon * cos(a))
Y = posY + (rayon * sin(a))
```

```
glBegin(GL_LINES);
glColor3f(1.0, 1.0, 1.0);
for (int a = 0; a < 360; a++)
{
    glVertex2f(posX + rayon *cos(a * PI / 180), posY + rayon *sin(a * PI / 180));
}
glEnd();
```

C1 :

```
posX = -1
posY = 1
rayon = 2
```



Boucles sur 360 points, ils se rejoignent pour former un cercle

d ← (2*π) / 360

Xi+1 = cos(d) * Xi – sin(d) * Yi

Yi+1 = cos(d) * Xi + cos(d) * Yi

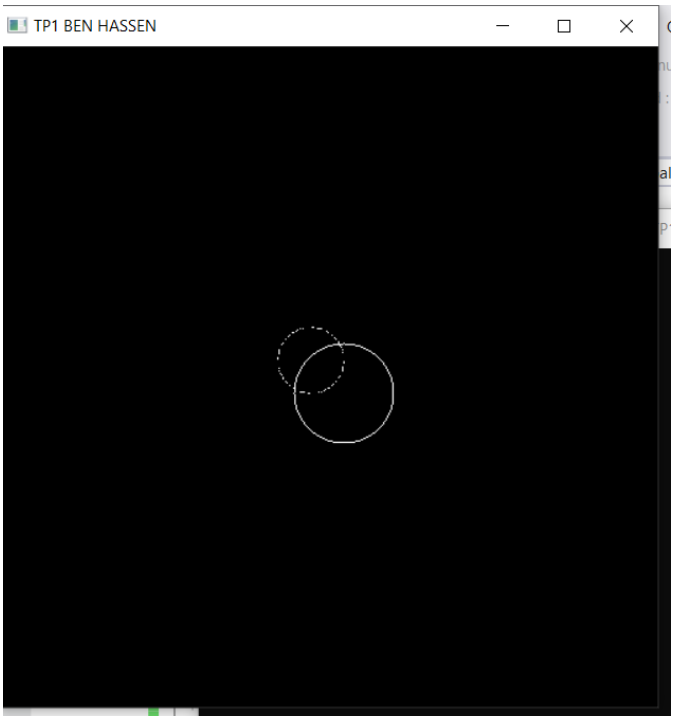
```
glBegin(GL_LINE_LOOP);
glColor3f(1.0, 1.0, 1.0);
for (int a = 0; a < 360; a++)
{
    xip = cos(d) * xi - sin(d) * yi;
    yip = sin(d) * xi + cos(d) * yi;
    glVertex3f(posX+xip, posY+yip, 0);
    xi = xip;
    yi = yip;
}
glEnd();
```

C2 :

posX = 1

posY = -1

rayon = 3



Pareil mais de façon incrémentale, chaque point est placé à Mi + 1

Exercice 10 :

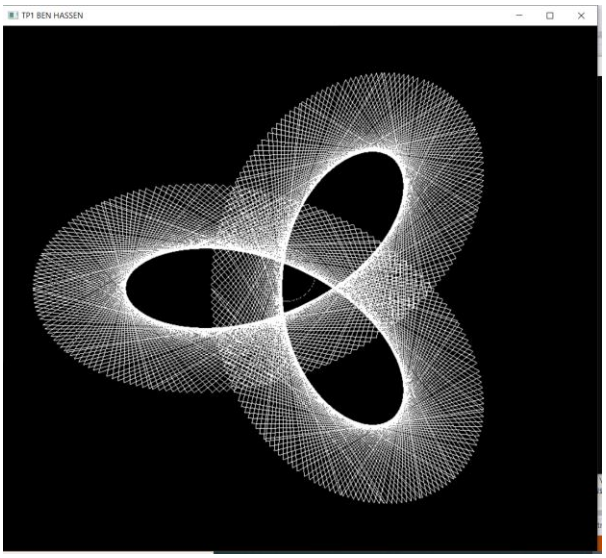
```
3*(x + iy)(1 - (x + iy))
(3x + 3iy)(1 – x – iy)
3x - 3x^2 – 3ixy + 3iy - 3iyx – 3i^2y^2
3x – 3x^2 - 3ixy + 3iy - 3iyx + 3y^2
3*(-x^2 + y^2 + x) – 3i*(2xy + y)

3 * ((-(x * x)) + (y * y) + x)
-3 * ((2 * x * y) + y)
```

```
    glBegin(GL_LINE_LOOP);
    glColor3f(1.0, 1.0, 1.0);
    for (int a = 0; a < 360; a++)
    {
        //-----
        xi = 2 * cos(a);
        yi = 2 * sin(a);

        xip = 3 * ((-(xi * xi)) + (yi * yi) + xi);
        yip = -3 * ((2 * xi * yi) + yi);

        glVertex2f(xip, yip);
    }
    glEnd();
```


TP1 BEN HASSEN