

```
#Importer les modules nécessaires
import Part
from FreeCAD import Base
import math
from math import sqrt
#Declarer les variables
x=0;y=0;z=0;i=0;j=0;k=0;Xi=0;Yi=0;Zi=0
#Obtenir le chemin du document actif
chemin1=str(Gui.ActiveDocument.Document.FileName)
#Ouvrir le fichier G-CODE on changeons l'extension dans le même chemin
chemin2=chemin1.replace("FCStd", "txt")
file=open(chemin2, "r")
#Parcourir chaque ligne du fichier
for bloc in file:
#Diviser chaque ligne a un ensemble des mots
    for mot in bloc.split():
#Vérifier qu'une coordonnée existe dans le mot et sauvegarder la valeur de cette coordonnée
        if 'X' in mot:
            x =mot[1:]
        if 'Y' in mot:
            y =mot[1:]
        if 'Z' in mot:
            z =mot[1:]
        if 'I' in mot:
            i =mot[1:]
        if 'J' in mot:
            j =mot[1:]
        if 'K' in mot:
            k =mot[1:]
#Vérifier que G0 existe dans la ligne
    if 'G0' in bloc.split():
#Définir les coordonnées du point de départ du déplacement suivant
        Xi=x; Yi=y; Zi=z
```

#Vérifier que G1 existe dans la ligne

```
elif 'G1' in bloc.split():
```

#Créer une ligne

```
ligne=Part.makeLine((Xi,Yi,Zi),(x, y, z))
```

#Mise en forme a l'écran

```
Part.show(ligne)
```

#Définir les coordonnées du point de départ du déplacement suivant

```
Xi=x; Yi=y; Zi=z
```

#Vérifier que G2 existe dans la ligne

```
elif 'G2' in bloc.split():
```

#Créer un arc

```
centre = Base.Vector(float(i),float(j),float(k))
point_depart = Base.Vector(float(Xi),float(Yi),float(Zi))
point_arrivee = Base.Vector(float(x),float(y),float(z))
point_ajoutee_H = Base.Vector(float(i)+15,float(j),float(k))
point_ajoutee_V = Base.Vector(float(i),float(j)+15,float(k))
Vecteur1 = centre.sub(point_depart)
Vecteur2 = centre.sub(point_arrivee)
Vecteur3 = centre.sub(point_ajoutee_H)
Vecteur4 = centre.sub(point_ajoutee_V)
ang1 = Vecteur1.getAngle(Vecteur3)
angle1 = math.degrees(ang1)
ang2 = Vecteur2.getAngle(Vecteur3)
angle2 = math.degrees(ang2)
ang3 = Vecteur1.getAngle(Vecteur4)
angle3 = math.degrees(ang3)
ang4 = Vecteur2.getAngle(Vecteur4)
angle4 = math.degrees(ang4)
rxj =( float(Xi) - float(i))**2
ryj =( float(Yi) - float(j))**2
Rayon = sqrt(rxj + ryj)
if float(angle4) <=90:
    if float(angle3) <= 90:
```

```
arc1 = Part.makeCircle(float(Rayon), Base.Vector(float(i),float(j),float(k)),
Base.Vector(0,0,1),float(angle2),float(angle1))
```

```
else:
```

```
arc1 = Part.makeCircle(float(Rayon), Base.Vector(float(i),float(j),float(k)),
Base.Vector(0,0,1),float(angle2),360-float(angle1))
```

```
else:
```

```
if float(angle3) <= 90:
```

```
arc1 = Part.makeCircle(float(Rayon), Base.Vector(float(i),float(j),float(k)),
Base.Vector(0,0,1),360-float(angle2),float(angle1))
```

```
else:
```

```
arc1 = Part.makeCircle(float(Rayon), Base.Vector(float(i),float(j),float(k)),
Base.Vector(0,0,1),360-float(angle2),360-float(angle1))
```

```
#Afficher l'arc
```

```
Part.show(arc1)
```

```
#Définir les coordonnées du point de départ du déplacement suivant
```

```
Xi=x; Yi=y; Zi=z
```

```
#Verifier que G3 existe dans la ligne
```

```
elif 'G3' in bloc.split():
```

```
#Dessiner un arc
```

```
centre = Base.Vector(float(i),float(j),float(k))
```

```
point_depart = Base.Vector(float(Xi),float(Yi),float(Zi))
```

```
point_arrivee = Base.Vector(float(x),float(y),float(z))
```

```
point_ajoutee_H = Base.Vector(float(i)+15,float(j),float(k))
```

```
point_ajoutee_V = Base.Vector(float(i),float(j)+15,float(k))
```

```
Vecteur1 = centre.sub(point_depart)
```

```
Vecteur2 = centre.sub(point_arrivee)
```

```
Vecteur3 = centre.sub(point_ajoutee_H)
```

```
Vecteur4 = centre.sub(point_ajoutee_V)
```

```
ang1 = Vecteur1.getAngle(Vecteur3)
```

```
angle1 = math.degrees(ang1)
```

```
ang2 = Vecteur2.getAngle(Vecteur3)
```

```
angle2 = math.degrees(ang2)
```

```
ang3 = Vecteur1.getAngle(Vecteur4)
```

```
angle3 = math.degrees(ang3)
```

```
ang4 = Vecteur2.getAngle(Vecteur4)
angle4 = math.degrees(ang4)
rxj =( float(Xi) - float(i))**2
ryj =( float(Yi) - float(j))**2
Rayon = sqrt(rxj + ryj)
if float(angle3) <=90:
    if float(angle4) <=90:
        arc2 = Part.makeCircle(float(Rayon), Base.Vector(float(i),float(j),float(z)),
Base.Vector(0,0,1),float(angle1),float(angle2))
    else:
        arc2 = Part.makeCircle(float(Rayon), Base.Vector(float(i),float(j),float(z)),
Base.Vector(0,0,1),float(angle1),360-float(angle2))
    else:
        if float(angle4) <=90:
            arc2 = Part.makeCircle(float(Rayon), Base.Vector(float(i),float(j),float(z)),
Base.Vector(0,0,1),360-float(angle1),float(angle2))
        else:
            arc2 = Part.makeCircle(float(Rayon), Base.Vector(float(i),float(j),float(z)),
Base.Vector(0,0,1),360-float(angle1),360-float(angle2))
#Afficher l'arc
Part.show(arc2)
#Définir les coordonnées du point de départ du déplacement suivant
Xi=x; Yi=y; Zi=z
#Afficher l'ensemble du contenu a l'ecran
Gui.SendMsgToActiveView("ViewFit")
#Placer l'ensemble en vue axonométrique
Gui.activeDocument().activeView().viewAxonometric()
```