

```
from random import seed
from random import randint
import math
def createBackground(maxWidth,maxHeight,r,g,b):
    arrAll=[]
    arrRow=[]
    colourByte=[]
    red=r
    green=g
    blue=b
    for intRows in range(maxHeight):
        colourByte.append(red)
        colourByte.append(green)
        colourByte.append(blue)
        for intCols in range(maxWidth):
            arrRow.append(colourByte)
        arrAll.append(arrRow)
        arrRow=[]
        colourByte = []
        if intRows < 255:
            green += 1
        elif intRows >= 255 and intRows <511:
            if intRows == 255:
                red = 255
                blue = 0
            else:
                blue += 1
```

```

        red -= 1

elif intRows >= 512 and intRows < 764:

    green -=1

else:

    blue -=1

    arrRow=[]

    colourByte = []

return arrAll

def drawCircle(arrPage, centreX, centreY, radius, r, g, b):

    colourByte=[]

    colourByte.append(r)

    colourByte.append(g)

    colourByte.append(b)

    for x in range(radius * -1, radius + 1):

        for y in range(radius * -1, radius + 1):

            if radius > round(math.sqrt(x**2+y**2)) :

                arrPage[int(x)+centreX][int(y+centreY)] = colourByte

    return arrPage

def drawCircleRev(arrPage, centreX, centreY, radius):

    for x in range(radius * -1, radius + 1):

        for y in range(radius * -1, radius + 1):

            if radius > round(math.sqrt(x**2+y**2)) :

                paint=arrPage[x+centreX][y+centreY]

                colourByte=[]

                temp=255-paint[0]

                colourByte.append(temp)

```

```

    temp=255-paint[1]
    colourByte.append(temp)
    temp=255-paint[2]
    colourByte.append(temp)
    arrPage[int(x)+centreX][int(y+centreY)] = colourByte
return arrPage

```

```

def drawCircleBorder(arrPage, centreX, centreY, longRadius, shortRadius,r, g, b):

```

```

    colourByte=[]
    colourByte.append(r)
    colourByte.append(g)
    colourByte.append(b)
    for x in range(longRadius * -1, longRadius + 1):
        for y in range(longRadius * -1, longRadius + 1):
            lengthOfLine = round(math.sqrt(x**2+y**2))
            if longRadius >= lengthOfLine and shortRadius <= lengthOfLine:
                arrPage[int(x+centreX)][int(y+centreY)]=colourByte
return arrPage

```

```

def saveToFile(Page,NameOfFile,VNum,cols,rows):

```

```

    myfile=open(NameOfFile+str(VNum)+".ppm",'w')
    myfile.write('P3' + "\n")
    myfile.write(str(cols)+" "+str(rows)+"\n")
    myfile.write(str(255)+"\n")
    for row in Page:
        for cVal in row:

```

```
    for c in cVal:
        myfile.write(str(c)+' ')
    myfile.write('\n')
myfile.close()
```

```
def lightRays(pageName,width,r,g,b):
```

```
    colourByte=[]
    colourByte.append(r)
    colourByte.append(g)
    colourByte.append(b)
    radius=int(width/4)
    x1=0
    y1=0
    seed(1)
    for angle in range(0,360):
        radian=angle*math.pi/180
        lineLength=radius+randint(0,radius)
        x1=int(lineLength*math.cos(radian))
        y1=int(lineLength*math.sin(radian))
        x2 = x1 * -1
        y2 = y1 * -1
        #print(x1, y1, x2, y2)
        if x1 != x2:
            m = (y1 - y2)/(x1 - x2)
        if y1 == y2:
            for x in range(x1,x2):
                pageName[y1+400][x+400]=colourByte
```

```

elif x1 == x2:
    for y in range(y1, y2):
        pageName[y+400][x1+400]=colourByte
elif m <= 1 and m>= -1:
    for x in range(x1, x2):
        y = round(m*(x - x2) + y2)
        if y <800:
            pageName[y+400][x+400]=colourByte
else:
    for y in range(y1, y2):
        x = round((y - y2)/m + x2)
        pageName[y+400][x+400]=colourByte
return pageName

```

```
def main():
```

```
    strFileName="ColoursSunRays2_"
```

```
    intFileFrame=1
```

```
    intMaxCols=800
```

```
    intMaxRows=800
```

```
    for angle in range(0, 180, 3):
```

```
        arrPage=createBackground(intMaxCols,intMaxRows, 255,0,0)
```

```
        arrPage = drawCircleBorder(arrPage, int(intMaxCols/2),int(intMaxRows/2),398,370,60,60,160)
```

```
        arrPage = drawCircleBorder(arrPage, int(intMaxCols/2),int(intMaxRows/2),340,310,60,60,160)
```

```
        arrPage = drawCircleBorder(arrPage, int(intMaxCols/2),int(intMaxRows/2),280,250,60,60,160)
```

```
        arrPage = drawCircleBorder(arrPage, int(intMaxCols/2),int(intMaxRows/2),220,190,60,60,160)
```

```
arrPage = drawCircleBorder(arrPage, int(intMaxCols/2),int(intMaxRows/2),160,130,60,60,160)
```

```
arrPage = drawCircleBorder(arrPage, int(intMaxCols/2),int(intMaxRows/2),100,70,60,60,160)
```

```
arrPage = drawCircleRev(arrPage, 70, 70, 50)
```

```
arrPage = drawCircleRev(arrPage, 70, 730, 50)
```

```
arrPage = drawCircleRev(arrPage, 730, 730, 50)
```

```
arrPage = drawCircleRev(arrPage, 730, 70, 50)
```

```
radian = math.radians(angle)
```

```
xPos = round(325*math.cos(radian))
```

```
yPos = round(325*math.sin(radian))
```

```
arrPage = lightRays(arrPage, intMaxCols, 255, 255, 125)
```

```
arrPage = drawCircle(arrPage, xPos+400, yPos+400, 45, 32, 0, 223)
```

```
radian = math.radians(angle+45)
```

```
xPos = round(325*math.cos(radian))
```

```
yPos = round(325*math.sin(radian))
```

```
arrPage = drawCircleRev(arrPage, xPos+400, yPos+400, 45)
```

```
radian = math.radians(angle+90)
```

```
xPos = round(325*math.cos(radian))
```

```
yPos = round(325*math.sin(radian))
```

```
arrPage = drawCircle(arrPage, xPos+400, yPos+400, 45, 32, 0, 223)
```

```
radian = math.radians(angle+135)
```

```
xPos = round(325*math.cos(radian))
```

```
yPos = round(325*math.sin(radian))
```

```
arrPage = drawCircleRev(arrPage, xPos+400, yPos+400, 45)
```

```
radian = math.radians(angle+180)
```

```
xPos = round(325*math.cos(radian))
```

```
yPos = round(325*math.sin(radian))
```

```
arrPage = drawCircle(arrPage, xPos+400, yPos+400, 45, 32, 0, 223)
```

```
radian = math.radians(angle+225)
```

```
xPos = round(325*math.cos(radian))
```

```
yPos = round(325*math.sin(radian))
```

```
arrPage = drawCircleRev(arrPage, xPos+400, yPos+400, 45)
```

```
radian = math.radians(angle+270)
```

```
xPos = round(325*math.cos(radian))
```

```
yPos = round(325*math.sin(radian))
```

```
arrPage = drawCircle(arrPage, xPos+400, yPos+400, 45, 32, 0, 223)
```

```
radian = math.radians(angle+315)
```

```
xPos = round(325*math.cos(radian))
```

```
yPos = round(325*math.sin(radian))
```

```
arrPage = drawCircleRev(arrPage, xPos+400, yPos+400, 45)
```

```
angle1 = angle+ 22.5
```

```
radian = math.radians(angle1)
```

```
xPos = round(206*math.cos(radian))
```

```
yPos = round(206*math.sin(radian))
```

```
arrPage = drawCircleRev(arrPage, xPos+400, yPos+400, 45)
```

```
radian = math.radians(angle1+45)
xPos = round(206*math.cos(radian))
yPos = round(206*math.sin(radian))
arrPage = drawCircle(arrPage, xPos+400, yPos+400, 45, 104, 0, 116)
```

```
radian = math.radians(angle1+90)
xPos = round(206*math.cos(radian))
yPos = round(206*math.sin(radian))
arrPage = drawCircleRev(arrPage, xPos+400, yPos+400, 45)
```

```
radian = math.radians(angle1+135)
xPos = round(206*math.cos(radian))
yPos = round(206*math.sin(radian))
arrPage = drawCircle(arrPage, xPos+400, yPos+400, 45, 104, 0, 116)
```

```
radian = math.radians(angle1+180)
xPos = round(206*math.cos(radian))
yPos = round(206*math.sin(radian))
arrPage = drawCircleRev(arrPage, xPos+400, yPos+400, 45)
```

```
radian = math.radians(angle1+225)
xPos = round(206*math.cos(radian))
yPos = round(206*math.sin(radian))
arrPage = drawCircle(arrPage, xPos+400, yPos+400, 45, 104, 0, 116)
```



```
radian = math.radians(angle1+270)
xPos = round(206*math.cos(radian))
yPos = round(206*math.sin(radian))
arrPage = drawCircleRev(arrPage, xPos+400, yPos+400, 45)

radian = math.radians(angle1+315)
xPos = round(206*math.cos(radian))
yPos = round(206*math.sin(radian))
arrPage = drawCircle(arrPage, xPos+400, yPos+400, 45, 104, 0, 116)

arrPage = drawCircleRev(arrPage, int(intMaxCols/2), int(intMaxRows/2), 45)
saveToFile(arrPage,strFileName,intFileFrame, intMaxRows, intMaxCols)
intFileFrame+=1
arrPage=[]
if __name__ == "__main__":
    main()

print("Programme finished")
```