```
import math
def createBackground(maxWidth,maxHeight):
    #initialises image to white
    arrAll=[]
    arrRow=[]
    for intRows in range(maxWidth):
        for intCols in range(maxWidth):
            arrRow.append(0)
        arrAll.append(arrRow)
        arrRow=[]
    return arrAll
def scaleShape(currentSize,maxRows, maxCols):

    x1 = currentSize
    y1 = round(currentSize/8)
    x3 = -1*x1
    y3 = -1*y1
    x2 = x1 * -1
    y2 = y1
    x4 = -1*x2
    y4 = -1*y2

    x1 += (maxRows)
    y1 += (maxCols)
    x2 += (maxRows)
    y2 += (maxCols)
    x3 += (maxRows)
```

```
    y3 += (maxCols)
    x4 += (maxRows)
    y4 += (maxCols)

    shape = []
    coord = []
    coord.append(round(x1))
    coord.append(round(y1))
    shape.append(coord)
    coord = []
    coord.append(round(x2))
    coord.append(round(y2))
    shape.append(coord)
    coord = []
    coord.append(round(x3))
    coord.append(round(y3))
    shape.append(coord)
    coord = []
    coord.append(round(x4))
    coord.append(round(y4))
    shape.append(coord)
    print(shape)
    return shape

def fillShape(arr, shape, maxRows, maxCols):
    x1 = shape[0][0]
    y1 = shape[0][1]
```

```
    x2 = shape[1][0]
    y2 = shape[1][1]
    x3 = shape[2][0]
    y3 = shape[2][1]
    x4 = shape[3][0]
    y4 = shape[3][1]
    AreaOfQuadrilateral = areaOfQuadrilateral(x1, y1, x2, y2, x3, y3, x4, y4)

    for arrRows in range(maxRows):
        AreaOfTriangles = 0
        for arrCols in range(maxCols):
            AreaOfTriangles += areaOfTriangle(arrRows, arrCols, x1, y1, x2,y2)
            AreaOfTriangles += areaOfTriangle(arrRows, arrCols, x2, y2, x3,y3)
            AreaOfTriangles += areaOfTriangle(arrRows, arrCols, x3, y3, x4,y4)
            AreaOfTriangles += areaOfTriangle(arrRows, arrCols, x1, y1, x4,y4)
            if AreaOfQuadrilateral == AreaOfTriangles:
                arr[arrCols][arrRows]=1
            AreaOfTriangles = 0
    return arr

def areaOfTriangle(x1, y1, x2, y2, x3, y3):
    area = abs((x1 *(y2 - y3)+ x2*(y3-y1)+x3*(y1-y2))/2)
    return area

def areaOfQuadrilateral(x1, y1, x2, y2, x3, y3, x4, y4):
    area = abs((x1 *(y2 - y3)+ x2*(y3-y1)+x3*(y1-y2))/2) +abs((x1 * (y4 - y3) + x4 * (y3 - y1) + x3 * (y1 -
y4))/2)
    return area
```

```
def saveFile(arrAll, maxRows, maxCols,fileName,versionNum):
    myfile=open(fileName+str(versionNum)+".pbm",'w')
    myfile.write('P1' +"\n")
    myfile.write(str(maxRows)+" "+str(maxCols)+"\n")
    for intRows in range(maxRows):
        myfile.write(getArray(arrAll[intRows])+"\n")
    myfile.close()


def getArray(passedValue):
    strOutString=""
    for intVal in passedValue:
        strOutString=strOutString+str(intVal)
    return strOutString


def main():
    arrPage = []
    strFileName="Scale"
    intVersionNumber = 0
    intMaxCols=800
    intMaxRows=800
    for size in range(20, 381,6):
        arrPage = createBackground(intMaxCols,intMaxRows)
        arrPage = fillShape(arrPage, scaleShape(size,intMaxCols/2,intMaxRows/2), intMaxRows,
intMaxCols)
        saveFile(arrPage,intMaxRows, intMaxCols,strFileName, intVersionNumber)
        intVersionNumber+=1
        arrPage=[]
```

```
if __name__ == "__main__":
    main()


print("Programme finished")
```