```
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 drawLine(arr, x1, y1, x2, y2):
    if x1 != x2:
        m = (y1 - y2)/(x1 - x2 )
    if y1 == y2:
        startX = min(x1, x2)
        endX = max(x1, x2)
        for x in range(startX,endX):
            arr[y1][x]=1
    elif x1 == x2:
        startY = min(y1, y2)
        endY = max(y1, y2)
        for y in range(startY, endY):
            arr[y][x1]=1
    elif m <= 1 and m>= -1:
        print("Gentle")
        if x1 < x2:
```

```
            for x in range(x1, x2):
                y = round(m*(x - x2) + y2)
                if y <800:
                    arr[y][x]=1
            print("Normal")
        else:
            for x in range(x2, x1):
                y = round(m*(x - x2) + y2)
                if y <800:
                    arr[y][x]=1
            print("Reverse")
    else:
        if y1 < y2:
            for y in range(y1, y2):
                x = round((y - y2)/m + x2)
                arr[y][x]=1
            print("Normal")
        else:
            for y in range(y2, y1):
                x = round((y - y2)/m + x2)
                arr[y][x]=1
            print("Reverse")
    return arr


def drawShape(arr, shape):
    for intCoords in range(len(shape)-1):
        startX = shape[intCoords][0]
```

```
        startY = shape[intCoords][1]
        endX = shape[intCoords+1][0]
        endY = shape[intCoords+1][1]
        arr=drawLine(arr,startX, startY, endX, endY)
    return arr

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):
    myfile=open(fileName+".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="Filling1"
```

```
    intMaxCols=800
    intMaxRows=800
    arrPage = createBackground(intMaxCols,intMaxRows)
    arrPage = fillShape(arrPage, [[150,350],[250,550],[550,450],[550,300]], intMaxRows, intMaxCols)
    saveFile(arrPage,intMaxRows, intMaxCols,strFileName)


if __name__ == "__main__":
    main()


print("Programme finished")
```