```
function DEADEND(N, x, y)
    for m=0 to xlength do
        neighbourcount ← 0, count ← 0
        for i=0 to xlength do
            a ← x[i] − x[m]
            b ← y[i] − y[m]
            if a == 0 and b == 0 then
                continue
            end if
            pointradius ← √(a² + b²)
            Θ ← atan(b/a)
            Θ ← Θ ∗ (180/3.145926)
            if a < 0 and b > 0 then
                Θ ← 180 − Θ
            end if
            if a < 0 and b < 0 then
                Θ ← 180 + Θ
            end if
            if a > 0 and b < 0 then
                Θ ← 360 − Θ
            end if
            updatedAnglesList[i] ← angle
            if pointradius <= r then
                neighbourcount + +
                for k=0 to 360 do
                    if Θ >= k and Θ < k + deg then
                        j ← ((k + deg)/deg) − 1
                        hash[j] + +
                    end if
                end for
            end if
        end for
        for w=0 to hashlength do
            if hash[w] == 0 then
                count + + ; ind[count − 1] ← w
            end if
        end for
        if count == 0 then
            NoDeadend
        else
            Deadend
            for q=0 to indlength do
                Call SelectPoint(ind[q])
            end for
        end if
        Call QuadrantPoints()
    end for
end function
```