```
clc;
close all;
clear all;

% Variables Intializition

maxWidth = 52;
maxHeight = 20;
referencePoint = 0.5;  % it'ii be used as a reference point in fingerprinting
APTransimissionPower = -10 ; % power in dB
PAF = 3;  %  Partition Attenuation Factor in dB
fc = 2.4e9 ; % carrier frequency 2.4 GHz
lamda = (3e8)/fc;
n = 3 ; % pathloss Exponent


% positions of Access points
AP1 = [6,15.5625];
AP2 = [17.5,4];
AP3 = [25.5,15.5625];
AP4 = [33.5,4];
AP5 = [45,15.5625];
APs = [AP1;AP2;AP3;AP4;AP5];

% we will creat a fingerPrint matrix with number of Columns equal to 5 (Number
% of Access Points) and Number of Rows represent how many grids we have.
fingerPriniting = zeros( maxHeight/referencePoint , maxWidth/referencePoint , length ↙
(APs));

% positions of walls
Walls= [0,0,6,15;6,0,8,15;14,0,8,8;14,8,8,7;22,0,3,15;25,0,2,15;
    27,0,3,15;30,0,8,8;30,8,8,7;38,0,8,15;46,0,6,15;0,15,2,5;
    2,16.5,4,3.5;6,16.5,4,3.5;10,16.5,4,3.5; 14,16.5,4,3.5;
    18,16.5,4,3.5;22,16.5,4,3.5;26,16.5,4,3.5;30,16.5,4,3.5;
    34,16.5 ,4,3.5;38,16.5,4,3.5;42,16.5,4,3.5];

[rows , cols ] = size(Walls);
% draw walls
drawWalls(Walls);

% to draw the Walls and the Access points on the same figure
hold on


% draw Access Points
drawAccessPoints(APs);


for row = 1 : maxHeight/referencePoint
```

```
    yGrid = row * referencePoint;  % y Position of the grid
    for col = 1 : maxWidth/referencePoint
        xGrid = col * referencePoint;  % X Position of the grid
        for ap = 1 : length(APs)
            xAP = APs(ap,1);  % X position of the ith AP
            yAP = APs(ap,2);  % Y position of the ith AP
            xLOS = [xAP , xGrid];
            yLOS = [yAP , yGrid];
            distance = sqrt((yGrid - yAP).^2 + (xGrid - xAP).^2);
            intersections = zeros(1 , 30);
            int = 1;
            for wall = 1 : rows
                if (wall  >= 5 && wall <=7)
                    continue;
                end
                xlimit = [Walls(wall , 1) Walls(wall , 1)+Walls(wall , 3)];
                ylimit = [Walls(wall , 2) Walls(wall , 2)+Walls(wall , 4)];
                xbox = xlimit([1 1 2 2 1]);
                ybox = ylimit([1 2 2 1 1]);
                [xi,yi] = polyxpoly(xLOS,yLOS,xbox,ybox);
                for i = 1 : length(xi)
                    intersections(int) = xi(i);
                    int = int +1;
                end
            end
            pathLoss = 10 * n * log10((4*pi*distance)/lamda);
            powerReceived = APTransimissionPower - (length(unique(intersections))-1) ↙
* PAF - pathLoss;
            fingerPriniting(row , col , ap ) = powerReceived;
        end
    end
end

% taking user input
inputPower = zeros(1,5);
for reading = 1 : 5
    inputPower(reading) = input('Reading From Access Point ');
end

[targetRow , targetCol ] = getLocation(inputPower , maxHeight , maxWidth , ↙
referencePoint , fingerPriniting);

% plotting the user's location in our map
plot( targetCol* referencePoint ,targetRow * referencePoint , '*r');

figure()
subplot(2,3,6);
for ap = 1 : 5
    subplot(2,3,ap);
    contourf(fingerPriniting(:,:,ap))
end

function [targetRow , targetCol ] = getLocation(inputPower , maxHeight , maxWidth , ↙
referencePoint , fingerPrinting)
    targetRow = 0;
    targetCol = 0;
    minDistance = 1e9;
    for row = 1 : maxHeight/referencePoint
        for col = 1 : maxWidth/referencePoint
            currentDistance = 0;
            for ap =1 : 5
                currentDistance = currentDistance + (inputPower(ap) - fingerPrinting ↙
(row , col ,ap))^2;
            end
            if (currentDistance < minDistance)
                targetRow = row;
                targetCol = col;
                minDistance = currentDistance;
            end
        end
    end
end

function drawWalls(Walls)
    for i = 1 : length(Walls)
        if (i >=1 && i <=4 || (i>=8 && i<12))
            rectangle('position' , Walls(i,:), 'FaceColor','g');
        elseif (i == 5 || i == 7)
            rectangle('position' , Walls(i,:), 'FaceColor','y');
        elseif ( i >= 12 )
            rectangle('position' , Walls(i,:), 'FaceColor','c');
        else
            rectangle('position' , Walls(i,:));
        end
    end
end

function drawAccessPoints(APs)
    for i = 1 : length(APs)
        rectangle('position' , [APs(i,1) APs(i,2) 0.5 0.5 ] , 'FaceColor','k');
    end
end
```