```
clc;
close all;
clear all;

%also no of nodes.
iterations = 1000;

% First of all, let us define the initial and goal point locations
q_start = [10 15];
q_goal = [-10 -10];
node_Array = [q_start, 1];
alpha = 0.75; %max distance between the twi consecutive nodes

%Let us now define our obstacles
OBSTACLES = [2 3 2; 7 8 2; -1 -1 1.5; -8 -5 4.5; 4 -5 2; 8 -6 2; 15 0 3; -10 8 2; -15 7 2; -15
n_obstacles = size(OBSTACLES,1); % number of obstacles
figure(1);
theta = 0:pi/10:2*pi;

% fill (obstacles(1,3)*cos(theta)+2, obstacles(1,3)*sin(theta)+3, 'R)

for i=1:2:n_obstacles
    % circle(obstacles(i,1), obstacles(i,2), obstacles(i,3))
    obs1 = [OBSTACLES(i,3)*cos(theta)+ OBSTACLES(i,1);OBSTACLES(i,3)*sin(theta)+OBSTACLES(i,2)
    fill(OBSTACLES(i,3)*cos(theta)+ OBSTACLES(i,1),OBSTACLES(i,3)*sin(theta)+OBSTACLES(i,2), 'c
    hold on;
    grid on;
    grid minor
end

xlabel('$\textit{\textbf{x-axis}}$','Interpreter','latex','FontSize',10)
ylabel('$\textit{\textbf{y-axis}}$','Interpreter','latex','FontSize',10)
title('$\textbf{Mohammed sohaib; 200611 -RRT}$','Interpreter','latex','FontSize',15);

for i = 2:2:n_obstacles
    %circle(obstacles(i,1), obstacles(i,2), obstacles(i,3));
    obs2 = [OBSTACLES(i,3)*cos(theta)+OBSTACLES(i,1);OBSTACLES(i,3)*sin(theta)+OBSTACLES(i,2)];
    fill(OBSTACLES(i,3)*cos(theta)+OBSTACLES(i,1),OBSTACLES(i,3)*sin(theta)+OBSTACLES(i,2),'m',
    hold on;
end

plot(q_start(1),q_start(2),'R*','MarkerSize',8);
text(q_start(1),q_start(2), 'Start',Color='red');
plot(q_goal(1),q_goal(2),'B*','MarkerSize',8);
text(q_goal(1),q_goal(2),'Goal',Color ='red');

for i = 1:iterations-1
    %consider a random point(x,y) in space and measure which point is
    %closest to this random point
```

```
    RandomPoint = 10*randn([1 2]);
    [SmallestArray] = ClosePoint(node_Array,RandomPoint);
    %this computes closest point form random point to the ccurrent array of
    %the start point
    Near_node = node_Array(SmallestArray,1:end-1);

    %new point is propostional to the distance from the nearest point
    New_node = Near_node + (RandomPoint-Near_node)/norm(RandomPoint-Near_node)*alpha;

    plot(Near_node(1),Near_node(2),'g.');
    %Distance from obstacle calculated by distance of new node from the
    %center of obstacle
    DistanceFromObstacle = ((New_node(1)-OBSTACLES(:,1)).^2+(New_node(2)-OBSTACLES(:,2)).^2).^(

    if sum(DistanceFromObstacle > OBSTACLES(:,end)) == n_obstacles
       % the above one checks if distance is greater for all the obstacles
       node_Array = [node_Array; [New_node, SmallestArray]];
       % bellow conection between two nodes
       NodeToNode = [Near_node; New_node];
       %Drawing line to connect nodes

       line(NodeToNode(:,1), NodeToNode(:,2),'Color', 'black','linewidth',2)
       %for animation
       drawnow limitrate;
       xlim([-20 20]);
       ylim([-20 20]);
       daspect([1 1 1])


    end

end

% Now we check whether one of our nodes is close to the goal point
[SmallestArray] = ClosePoint(node_Array,q_goal);
%nearest point to the goal
Near_node = node_Array(SmallestArray,1:end-1);
NodeToNode = [Near_node; q_goal];


line(NodeToNode(:,1),NodeToNode(:,2),'Color', 'B','linewidth',2);


% if so, then select path connecting start point and the goal point

while ( Near_node ~= q_start)

    ParentArray = node_Array(SmallestArray,end);
    ParentNode = node_Array(ParentArray,1:end-1);
    NodeToNode = [Near_node; ParentNode];
```

```
    line(NodeToNode(:,1),NodeToNode(:,2),'Color','B','linewidth',2)

    SmallestArray = ParentArray;
    Near_node = ParentNode;
    pause(0.01);
    daspect([1 1 1])

end
```