```
clear, clc
x = [4,5,1];grad = @func_deriv;ep = 1e-6;
x_star = GD_secant(grad,x,ep);
disp(func(x_star))
```

```
   2.3797e-10
```

```
x = [-2,2];grad = @Rosenbrock_grad; ep = 1e-4;
x_star = GD_secant(grad,x,ep);
disp(x_star)
```

```
    0.9999    0.9998
```

```
function [x_new] = linesearch_secant(grad,x,d)
% This function computes the line search on direction d with secant method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input
% grad: gradient function
% x: starting point
% d: gradient decent direction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output
% x_new: minimizer of the function on the decent direction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xa = x; xb = xa + d'/norm(d); d_norm = norm(d); % Initialization
for i = 1:100
fda = d'*(grad(xa)*d)/d_norm; % Function gradient of xa projected on the decent direct
fdb = d'*(grad(xb)*d)/d_norm; % Function gradient of xb projected on the decent direct
alpha = norm(xb-xa)/(norm(fdb)-norm(fda)); x_new = xb - alpha*fdb; % Compute the x of
xa = xb; xb = x_new; % Update virable values
if abs(grad(x_new)*d) <= 0.01* abs(grad(x)*d)
    break
end
end
end

function [x] = GD_secant(grad,x,ep)
% This function computes the gradient decent with secant method for 5000
% steps or untill the gradient decreases to ep
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input
% grad: gradient function
% x: starting location x
% ep: termination thereshold of gradient
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output
% x: minimizer according to secant method
for i = 1:5000
    d = -grad(x)'; % update d to the negative of gradient vector
```

```
    x = linesearch_secant(grad,x,d); % compute the minimizer of line search on directi
    if norm(d) <= ep
        break
    end
end
end


function [fd] = Rosenbrock_grad(x)
fd = zeros(1,2);
fd(1) = 400*x(1)^3 + (2-400*x(2))*x(1)-2;
fd(2) = 200*(x(2)-x(1)^2);
end

function [fd] = func_deriv(x)
fd = zeros(1,3);
fd(1) = 4*(x(1)-4)^3;
fd(2) = 2*(x(2)-3);
fd(3) = 16*(x(3)+5)^3;
end
```