```
%% EXERCISE 1

% ----------------------------------------
% Point a
% ----------------------------------------

function [res, err] = lagr_polynomial(x)

f=@(x) cos(x);

x_nodes=[0,0.6,0.9];

pp=polyfit(x_nodes,f(x_nodes),2);
res=polyval(pp,x);
err = abs(res-f(x));

fprintf('The values is %10.4f \n', res)
fprintf('The error is %10.4f \n', err)


xx=linspace(0,1,1001);
figure()
hold on
for k=1:3
    plot(xx,lagr_poly(x_nodes,k,xx))
    plot(x_nodes(k),lagr_poly(x_nodes,k,x_nodes(k)),'s')
end
end

% ----------------------------------------
function y = lagr_poly(nodes, k, x)
n=length(nodes);

y=ones(size(x));
for z=1:n
    if z ~=k
        y=y.*(x-nodes(z))/(nodes(k)-nodes(z));
    end
end

end
% ----------------------------------------


% ----------------------------------------
% Point b
% ----------------------------------------
f=@(x) cos(x);
n=2;

f3=@(x) sin(x);

x_nodes=[0,0.6,0.9];

xx=linspace(0,1,10000);

boundf3 = max(abs(f3(xx)));
boundpoly = max(abs(poly_nodes(x_nodes,xx)));
fprintf('bound for f3 is %f \n', boundf3)
fprintf('bound for poly is %f \n' , boundpoly)
fprintf('bound for product is %f \n', boundf3*boundpoly/6)

figure()
hold on
plot(xx, abs(f3(xx)))
plot(xx,abs(poly_nodes(x_nodes,xx)))
legend('f3','poly')
```

```
% ----------------------------------------
function y= poly_nodes (nodes, x)
n=length(nodes);
y=ones(size(x));

for k=1:n
    y=y.*(x-nodes(k));
end

end
% ----------------------------------------
```

```
%% EXERCISE 2

% ----------------------------------------
% Point a
% ----------------------------------------

f=@(x) exp(-x.^2);
d=5;

degs=2:2:10;
x_test = -d:2*d/1000:d;
plot(x_test,f(x_test))
hold on
for k =1:length(degs)
    deg=degs(k);
    nodes=-d:2*d/deg:d;
    pp=polyfit(nodes,f(nodes),deg);
    y_test= polyval(pp,x_test);
    plot(x_test,y_test)
end
legend('exact','2','4','6','8','10')
savefig('interp_n.fig')


% ----------------------------------------
% Point b
% ----------------------------------------
f=@(x) exp(-x.^2);
d=5;
x_nodes = linspace(-d,d,9);
x_test = -d:2*d/1000:d;
pwlinear=piecewise_linear(x_nodes,f(x_nodes),x_test);
pwcubic=spline(x_nodes,f(x_nodes),x_test);
figure()
plot(x_test,f(x_test))
hold on
plot(x_test, pwlinear)
plot(x_test, pwcubic)
savefig('piecewise.fig')
```