```
function weights=fdweights(pts,D);
% Calculates Finite Difference Weights need to approximate the Dth
% derivative of a function at the points indicated by the vector pts.
% Inputs:
%   pts:    a vector containing the relative postion of the points from the
%           center of the finite difference scheme.
%   D:      the derivative of interest (0<D<length(pts)-1);
% Output:
%   weights: the finite difference weights that determine the derivative

z=length(pts);
if z<=D, %can only calculate derivatives up through z-1;
    display('Error: D must be less than the number of points used in
approximation')
    return;
end

% Build the Matrix
M=pts.^0;
for i=1:z-1
    M=[M;pts.^i];
end

%Build the right hand side vector;
V(1:length(pts))=0;
V(D+1)=factorial(D);
V=V';

%determine the FD Weights
weights=inv(M)*V;

%determine the error
e=sum(weights.*[pts.^z./factorial(z)]');
p=z-D;
if abs(e)<10*eps;
    e=sum(weights.*[pts.^(z+1)./factorial(z+1)]');
    p=p+1; D=D+1;z=z+1;
end

display(['error = ',num2str(rats(e)),'  h^',num2str(p),'
f^(',num2str(z),')(x)']);
weights=[num2str(pts'),rats(weights,20)],
```