```
function [yp,y1,y2, res, Sres] = parabola(x,y,xp)
% computes coefficients for best quadratic fit:
% y = a + b*x + c*x^2
% by solving the normal equations: X'*X*[a;b;c]=X'*y
%
m=length(x);
% define entries for X'*X, where X is the Design Matrix
X1 = sum(x);
X2 = sum(x.*x);
X3 = sum(x.^3);
X4 = sum(x.^4);
% define entries for rhs., X'*y
Y1 = sum(y);
XY = sum(x.*y);
X2Y= sum(x.*x.*y);
AA = [m X1 X2; X1 X2 X3; X2 X3 X4];
bb = [Y1; XY; X2Y];
% Find polynomial coefficients for least squares fit
xx= AA\bb;
a = xx(1);b = xx(2); c = xx(3);
% ZZ = (X'*X)^(-1)
ZZ = inv(AA);
% compute predicted y and y1 = yp+P, y2 = yp-P
yp = a + b*xp + c*xp.^2;
% compute residual vector
res = a + b*x + c*x.*x - y;
Sres = sum(res.*res);
% compute 5% confidence intervals (using formula (3.96) from Hocking):
%yp^(+/-) =  a^T * xp  +/- t(a/2; N-p)* sqrt{var(a^T * xp)}
sig  = Sres/(m-3);
k = length(xp);
for i = 1:k
   STDER2(i) = sig * [1 xp(i) xp(i)^2]*ZZ*[1 ;xp(i);xp(i)^2];
end
tee = stud(m-3,.025);
y1 = yp+tee*sqrt(STDER2);
y2 = yp-tee*sqrt(STDER2);
```