```
clear all;clc

x = 1:1:10;
y = log(x);

xp1 = 1.43;
xp2 = 5.78;
xp3 = 10.01;

yp1 = myfuninterpolation(x,y,xp1)
yp2 = myfuninterpolation(x,y,xp2)
yp3 = myfuninterpolation(x,y,xp3)



% I made a function as per the question
```

My Function declaration

```
function ypo = myfuninterpolation(x,y,xp)

 if ((xp > 2) && (xp < 9))  % Given that if point lies in interval but
 not at end points then do
                              %do cubic interpolation
     for i = 2 : length(x)-1                           % I am also
 asuming that sometimes the data you give is not sorted so doing both
 in if loop for minimum
         min = 100;
         if ((xp - x(i)) > 0)  && (xp - x(i)) < min
              min = (xp - x(i));
              start = i;
         end
     end
     start = start -1;
     for j = 1:4
         xintervl(j) = x(start);
         yintervl(j) = y(start);
         start = start+1;
     end

    ypo = lagrangeinterpolation(xintervl,yintervl,xp);

     %%do a cubic interpolation
 else                            % If it is at the end points and
 inside the given interval then do quadratic lagrange polynomial
    if xp < 2 && xp > x(1)
        start = 1;
        for j = 1:5
         xintervl(j) = x(start);
         yintervl(j) = y(start);
         start = start+1;
        end
     ypo = lagrangeinterpolation(xintervl,yintervl,xp);
    else
        if xp <= x(end) && xp > x(end-1)
        start = length(x)-4;
        for j = 1:5
         xintervl(j) = x(start);
         yintervl(j) = y(start);
         start = start+1;
        end
     ypo = lagrangeinterpolation(xintervl,yintervl,xp);

        else                  % If My point is outside the range of x
 then put a error message as per question  and assign NAN, can't find
 any better option than that
            ypo = nan;
        fprintf('\nERROR: %.2f is outside the Range of X\n',xp);
     %%do a quadratic lagrange polynomial\
        end
 end

 end
end

%%Lagrange interpolation Function
function ypl = lagrangeinterpolation(x,y,xp)
 n = length(x)-1;

 sum = 0;
 for i = 1:n+1
     pr = 1;
     for j = 1:n+1
         if j ~= i
             pr = pr * (xp-x(j))/(x(i)-x(j));
         end
     end
     sum = sum + y(i) *pr;
 end
 ypl = sum;
end
```

```
yp1 =

    0.3490


yp2 =

    1.7545


ERROR: 10.01 is outside the Range of X

yp3 =
    NaN
```

Published with MATLAB® R2018b