1. (10) Let
\[
s(x) = \begin{cases} 
(x - 2)^3 + 2(x - 2) + 1 & \text{if } x \leq 2 \\
-x^3 + 6x^2 - 10x + 5 & \text{if } x > 2 
\end{cases}
\]

Is \(s\) a cubic spline? Justify your answer.

**Answer:** We need to verify that \(s\) has degree at most 3 (obvious) and that \(s, s', \text{ and } s''\) are continuous. It is obvious that they are continuous everywhere except at \(x = 2\), so we check there.

\[
s'(x) = \begin{cases} 
s'_1(x) = 3(x - 2)^2 + 2 & \text{if } x \leq 2 \\
s'_2(x) = -3x^2 + 12x - 10 & \text{if } x > 2 
\end{cases}
\]

\[
s''(x) = \begin{cases} 
s''_1(x) = 6(x - 2) & \text{if } x \leq 2 \\
s''_2(x) = -6x + 12 & \text{if } x > 2 
\end{cases}
\]

Then we can see that \(s_1(2) = s_2(2) = 1, s'_1(2) = s'_2(2) = 2, \text{ and } s''_1(2) = s''_2(2) = 0\), so \(s\) is a cubic spline. (Also note that \(s''_1 \neq s''_2\), so \(s\) is truly a spline, not just a cubic polynomial.)

2. (10) (P3.1.1) Modify the Locate function so that it tries \(i = g + 1\) and \(i = g - 1\) before resorting to binary search. (Take care to guard against subscript out-of-range.)

```matlab
function i = Locate(x,z,g)
% i = Locate(x,z,g)
% Locates z in a partition x.
% x is column n-vector with x(1) < x(2) < ... < x(n) and
% z is a scalar with x(1) <= z <= x(n).
% g is an optional 3rd argument that satisfies 1 <= g <= n-1.
% i is an integer such that x(i) <= z <= x(i+1).
% Before the general search for i begins, the value i=g is tried.
% and we also try g-1 and g+1 if possible.

if nargin==3
% Try the initial guess.
```
if (x(g)<=z) & (z=x(g+1))
    i = g;
    return
end

if (g>1)
    if (x(g-1)<= z) & (z <= x(g))
        i=g-1;
        return
    end
end

if (g < length(x)-1)
    if (x(g+1)<= z) & (z <= x(g+2))
        i=g+1;
        return
    end
end

end

n = length(x);
if z=x(n)
    i = n-1;
else  % Binary Search
    Left = 1;
    Right = n;
    while Right > Left+1  % x(Left) <= z <= x(Right)
        mid = floor((Left+Right)/2);
        if z < x(mid)
            Right = mid;
        else
            Left = mid;
        end
    end
    i = Left;
end