```
% Given data points
x = [1, 2, 2.5, 3, 4, 5];
f_x = [0, 5, 7, 6.5, 2, 0];

% Point to interpolate
x_interp = 3.4;

% Initialize variables for interpolated values
f_interp_order1 = 0;
f_interp_order2 = 0;
f_interp_order3 = 0;

% Lagrange interpolation polynomial of order 1
for i = 1:length(x)
    L = 1;
    for j = 1:length(x)
        if j ~= i
            L = L * (x_interp - x(j)) / (x(i) - x(j));
        end
    end
    f_interp_order1 = f_interp_order1 + f_x(i) * L;
end

% Lagrange interpolation polynomial of order 2
for i = 1:length(x)
    L = 1;
    for j = 1:length(x)
        if j ~= i
            L = L * (x_interp - x(j)) / (x(i) - x(j));
        end
    end
    f_interp_order2 = f_interp_order2 + f_x(i) * L;
end

% Lagrange interpolation polynomial of order 3
for i = 1:length(x)
    L = 1;
    for j = 1:length(x)
        if j ~= i
            L = L * (x_interp - x(j)) / (x(i) - x(j));
        end
    end
    f_interp_order3 = f_interp_order3 + f_x(i) * L;
end

fprintf('f(3.4) using Lagrange interpolation of order 1: %.4f\n', f_interp_order1);
fprintf('f(3.4) using Lagrange interpolation of order 2: %.4f\n', f_interp_order2);
fprintf('f(3.4) using Lagrange interpolation of order 3: %.4f\n', f_interp_order3);
```

```
f(3.4) using Lagrange interpolation of order 1: 4.8248
f(3.4) using Lagrange interpolation of order 2: 4.8248
f(3.4) using Lagrange interpolation of order 3: 4.8248
```

Published with MATLAB® R2023b