```
% 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;

% Newton's interpolating polynomial of order 1
for i = 1:2
    f_interp_order1 = f_interp_order1 + f_x(i) * prod(x_interp - x(1:i))/(x(i+1) - x(i));
end

% Newton's interpolating polynomial of order 2
for i = 1:3
    f_interp_order2 = f_interp_order2 + f_interp_order1 + f_x(i) * prod(x_interp - x(1:i))/(x(i+1) - x(i));
end

% Newton's interpolating polynomial of order 3
for i = 1:4
    f_interp_order3 = f_interp_order3 + f_interp_order2 + f_x(i) * prod(x_interp - x(1:i))/(x(i+1) - x(i));
end

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

```
f(3.4) using Newton's interpolating polynomial of order 1: 33.6000
f(3.4) using Newton's interpolating polynomial of order 2: 176.7360
f(3.4) using Newton's interpolating polynomial of order 3: 790.7424
```

Published with MATLAB® R2023b