```
function [t, w_4step1, w_4step2] = AdamBashforthPredictorMethod2Ord(f1, f2, a, b, N,↙
alpha1, alpha2)

    % Define initial values and prepare for iteration
    h = (b - a)/N;
    t(1) = a;
    w1(1) = alpha1;
    w2(1) = alpha2;

    for i = 1:3

        % Using RK4, find the first 4 approximations of w1 and w2. These
        % will be used in the prediction-correction portion, which requires
        % the past 4 points
        K11 = h*f1(t(i), w1(i), w2(i));
        K12 = h*f2(t(i), w1(i), w2(i));
        K21 = h*f1(t(i) + h/2, w1(i) + K11/2, w2(i) + K12/2);
        K22 = h*f2(t(i) + h/2, w1(i) + K11/2, w2(i) + K12/2);
        K31 = h*f1(t(i) + h/2, w1(i) + K21/2, w2(i) + K22/2);
        K32 = h*f2(t(i) + h/2, w1(i) + K21/2, w2(i) + K22/2);
        K41 = h*f1(t(i) + h, w1(i) + K31, w2(i) + K32);
        K42 = h*f2(t(i) + h, w1(i) + K31, w2(i) + K32);

        w1(i+1) = w1(i) + (K11 + 2*K21 + 2*K31 + K41)/6;
        w2(i+1) = w2(i) + (K12 + 2*K22 + 2*K32 + K42)/6;
        t(i+1) = a + i*h;

    end

    w_4step1 = w1;
    w_4step2 = w2;

    for i = 4:N

        t(i+1) = a + i*h;

        % Implement the explicit 4th order method first to obtain the
        % initial guess, then use the implicit 4th order method to correct
        % the guess.
        w_temp1 = w_4step1(i) + h*(55*f1(t(i), w_4step1(i), w_4step2(i)) - 59*f1(t(i-1),↙
w_4step1(i-1), w_4step2(i-1)) + 37*f1(t(i-2), w_4step1(i-2), w_4step2(i-2)) - 9*f1(t(i-↙
3), w_4step1(i-3), w_4step2(i-3)))/24;
        w_temp2 = w_4step2(i) + h*(55*f2(t(i), w_4step1(i), w_4step2(i)) - 59*f2(t(i-1),↙
w_4step1(i-1), w_4step2(i-1)) + 37*f2(t(i-2), w_4step1(i-2), w_4step2(i-2)) - 9*f2(t(i-↙
3), w_4step1(i-3), w_4step2(i-3)))/24;
        w_4step1(i+1) = w_4step1(i) + h*(9*f1(t(i+1), w_temp1, w_temp2) + 19*f1(t(i),↙
w_4step1(i), w_4step2(i)) - 5*f1(t(i-1), w_4step1(i-1), w_4step2(i-1)) + f1(t(i-2),↙
w_4step1(i-2), w_4step2(i-2)))/24;
        w_4step2(i+1) = w_4step2(i) + h*(9*f2(t(i+1), w_temp1, w_temp2) + 19*f2(t(i),↙
w_4step1(i), w_4step2(i)) - 5*f2(t(i-1), w_4step1(i-1), w_4step2(i-1)) + f2(t(i-2),↙
w_4step1(i-2), w_4step2(i-2)))/24;
```

```
    end

end
```