```matlab
function x = ERKTemplate(ButcherArray, f, T, x0)
    % Returns the iterations of an ERK method
    % ButcherArray: Struct with the ERK's Butcher array
    %   A: Nstage x Nstage
    %   b: Nstage x 1
    %   c: Nstage x 1
    %       (NB: both b and c must be standing vectors)
    % f: Function handle
    %   Vector field of ODE, i.e., x_dot = f(t,x)
    % T: Vector of time points, 1 x Nt
    % x0: Initial state, Nx x 1
    % x: ERK iterations, Nx x Nt
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Define variables
    % Allocate space for iterations (x) and k1,k2,...,kNstage
    % It is recommended to allocate a matrix K for all kj, i.e.
    % K = [k1 k2 ... kNstage]
    A = ButcherArray.A;
    b = ButcherArray.b;
    c = ButcherArray.c;

    nx = size(x0,1);
    Nt = size(T,2);
    Nstage = size(A,1);

    K = zeros(nx,Nstage);
    x = zeros(nx,Nt);
    x(:,1) = x0;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    xt = x0; % initial iteration
    % Loop over time points
    for nt=2:Nt
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Update variables
        dt = T(nt) - T(nt-1);

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Calculation of the K vector relies on the A matrix having zeros
        % on and above the diagonal such that it's explicit RK.
        K(:,1) = f(nt, xt);

        % Loop that calculates k2,...,kNstage
        for nstage=2:Nstage
            K(:,nstage) = f(nt, xt + dt * sum(K .* A(nstage,:),2) );
        end % for
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Calculate and save next iteration value x_t
        xt = xt + dt * sum(K .* (b.'),2);
        x(:,nt) = xt;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    end % for
end % function
```