```
% c)

function [f] = FourierCoefficients(N, r, T, k, mu_w, sigma_w)
% Computes first N Fourier coefficients for European call via recursive formula

% N: number of first coefficients to compute
% r: parameter of the Jacobi model
% T: maturity time (starting from zero t=0)
% k: strike
% mu_w, sigma_w: mean ans standard deviation of the w(x) gaussian density


% vector initialization
f = zeros(N+1,1);

% probabilistic standard Hermite polynomial
H = @(n,x) 2^(-0.5*n) * hermiteH(n,x/sqrt(2));

C = (k - mu_w) / sigma_w;   % useful in next steps (first parameter in I)

% auxiliary recursive function initialization
I = exp(0.5 * sigma_w^2) * normcdf(sigma_w - C);

% first coefficient f0
f(1) = exp(-r * T + mu_w) * I - exp (-r * T + k) * normcdf(-C);

% Recursive part:
for n = 1:N

    % Fourier coefficients
    f(n+1) = exp(-r * T + mu_w) * sigma_w / sqrt(factorial(n)) * I;

    % ausiliary function
    I = H(n-1,C) * exp(sigma_w * C) * normpdf(C) + sigma_w * I;

    % note that we change I after f becuase to calculate fn we need In-1
end

end
```