```
function e = evalForney(omega, lambda_prime, X, GF)
%{
EVALFORNEY evaluates the forney expression for a given X error
 location
INPUTS:
    omega - the omega polynomial or the error evaluator polynomial in
 power
            form
    lambda_prime - the formal derivative of the lambda (error locator)
                   polynomial in power form
    X - the error location that the forney expression evaluates at to
        determine the error value
    GF - Array of cells mapping the binary groups to GF(2^m) alpha
 coeff
OUTPUT:
    e - the error value (power of alpha) to be added to the
 coefficient at
        the error location
%}

n = size(GF, 1) - 1; %needed for inversing values of X;

dvd = MultGF2(X, EvalPolyGF2(omega, n-X, GF), GF); %dividened
dvs = EvalPolyGF2(lambda_prime, n-X, GF); %divisor

e = DivGF2(dvd, dvs, GF);

end
```

Published with MATLAB® R2018b