```
function cur_prod = PolyMultGF2(A,B, GF)
%   -prod is the return value that gives the power form array of the
 product of
%    A and B. The elements of A and B are the powers of the exponents
 of
%    alpha coefficients (-1 == inf). Their placement represents the
 degree
%    of the xvalue (highest degree on left)
%   -A and B are power form matrix parameters that will be multiplied
 together
%   -GF (nx1 cell) is the power and rectangular form enumeration of
 the given
%   field, the indecies of the cell matrix represent power or pow = i
 - 2, (-1 == inf)
%   -n is the p^m big Galois Field
%Created By Brendan Cain
%Error Correcting Codes HW 7
m = size(GF{1},2);
n = 2^m;

a_len = size(A,2);
b_len = size(B,2);
cur_prod = zeros(1, a_len+b_len-1); %allocate space for return value
cur_prod(:) = -1; %put in terminator value (lets algorithm know this
 space is untouched)
for i = 1:a_len
    for j = 1:b_len
        if(A(1,i) ~= -1 && B(1,j) ~= -1) %only do math if not inf (inf
 times anything is inf)
            pow = i + j - 1; %gives the current place in the product
 array
            val = cur_prod(1, pow); %previous product that has same
 degree
            exp = mod((A(1,i) + B(1,j)),n-1); %current product of
 element multiplication
            if(val == -1) %check to see if touched
                cur_prod(1, pow) = exp;
            else
                %do rectagular form addition (match qith equiv cell)
                new_val = double(xor(GF{val+2}, GF{exp+2}));
                %gets index that the specific value is at
                exp = cellfun(@(x)isequal(x, new_val),GF, 'un', 0);
                exp = find([exp{:}] == 1) - 2; %subtracts two to get
 exp of alpha
                cur_prod(1, pow) = exp;
            end
        end
    end
end

end
```

Published with MATLAB® R2018b