```
% Homework 3

%% Question 2

a_1 = [1; 2; 0; 1];
a_2 = [0; -1; 3; 0];
a_3 = [2; 0; 1; -1];
b = [3; 0; -1; 2];

rref([a_1, a_2, a_3, b])

%% Question 3

v_1 = [1; 0; 2];
v_2 = [3; -1; 1];
v_3 = [2; -1; -1];
v_4 = [4; -1; 3];

% 3(a)
% Just augument and see if they are full rank.
rank([v_1, v_2, v_3, v_4])  % rank = 2, so no.

%% Question 4

% 4(a)
function B = myGS(A)
    [m, n] = size(A);
    if n > m
        error("`A` must be an m x n matrix where n <= m, got %d x %d", m, n)
    end
    if rank(A) < n
        error("Expecting rank(A) = %d, got rank(A) = %d", n, rank(A))
    end

    B = zeros(m, n);

    for i = 1:n
        w_i = A(:, i);          % column to orthogonalize
        v_i = w_i;              % to hold the orthogonalized
        for j = 1:i-1
            v_j = B(:,j);
            v_i = v_i - (dot(w_i, v_j) / dot(v_j, v_j) * v_j);
        end

        % Place the orthogonalized vector in B.
        B(:,i) = v_i;
    end

    % Normalize
    for i = 1:n
        B(:,i) = B(:,i) / norm(B(:,i));
    end
end

% 4(b)
disp("Using `myGS`. Normalizing afterwards.")
myGS([
    1 0 1 1;
    0 1 0 1;
```

very good...

```
    1 0 0 1;
    0 -1 1 1
])


% 4(c)
function B = myGS2(A)
    [m, n] = size(A);
    if n > m
        error("`A` must be an m x n matrix where n <= m, got %d x %d", m, n)
    end
    if rank(A) < n
        error("Expecting rank(A) = %d, got rank(A) = %d", n, rank(A))
    end

    B = zeros(m, n);
```



```
    for i = 1:n
        w_i = A(:, i);          % column to orthogonalize
        v_i = w_i;              % to hold the orthogonalized
        for j = 1:i-1
            v_j = B(:,j);
            v_i = v_i - (dot(w_i, v_j) / dot(v_j, v_j) * v_j);
        end

        % Place the orthonormalized vector in B.
        B(:,i) = v_i / norm(v_i);
    end
end

disp("Using `myGS2`. Normalizing on-the-fly.")
myGS2([
    1 0 1 1;
    0 1 0 1;
    1 0 0 1;
    0 -1 1 1
])
```

yes !