```
% Homework 2

format rational  % Display fractions where possible.

%% Question 1

% 1(a)
function U = myPartialPivot(A)
    [m, n] = size(A);
    if n ~= m
        error("`A` must be an n x n matrix, got: %d x %d", m, n)
    end

    % Keep track of row swaps, see Question 2(c).
    rowSwaps = 0;

    for j = 1:n  % for each column

        % Locate the row with biggest absolute value.
        maxAbsValue = 0;
        maxRowIndex = 0;
        for i = j:m  % for each row from pivot
            absValue = abs(A(i,j));
            if absValue > maxAbsValue
                maxAbsValue = absValue;
                maxRowIndex = i;
            end
        end

        % If everything is zero below, skip.
        if maxAbsValue == 0
            continue
        end

        % Perform swap, if necessary.
        if maxRowIndex ~= j
            pivotRow = A(j,:);
            A(j,:) = A(maxRowIndex, :);
            A(maxRowIndex, :) = pivotRow;
            rowSwaps = rowSwaps + 1;
        end

        % Eliminate rows.
        for i = j+1:m
            ratio = A(i,j) / A(j,j);
            A(i,:) = A(i,:) - ratio * A(j,:);
        end
    end

    U = A;
    fprintf("The number of row interchanges is %d.", rowSwaps)
end

% 1(b)
function r = myRank(A)
    [m, n] = size(A);
    if n ~= m
        error("`A` must be an n x n matrix, got: %d x %d", m, n)
    end
```

Good 4/4

```
    U = myPartialPivot(A);
    r = 0;
    for i = 1:n
        %%%%%% YOU MUST ROUND TO AVOID                    %%%%%%
        %%%%%% FLOATING-POINT ARITHMETIC ISSUES          %%%%%%
        %%%%%% AROUND 10 D.P. APPEARS TO BE ACCEPTABLE. %%%%%%
        %%%%%% SEE REPORT FOR MORE INFORMATION.          %%%%%%
        if round(U(i,i), 10) == 0
            break
        end
        r = r + 1;
    end
end

% 1(c)
P = rand(5, 3)
Q = rand(3, 5)
A = P * Q
rank(A)


%% Question 2

% 2(a)
A = [
    4 1 1 1;
    1 4 1 1;
    1 1 4 1;
    1 1 1 4;
];

% 2(c)
disp("Testing partial pivoting with the matrix:")
disp(A)
myPartialPivot(A)

% Testing with swaps
B = [
    1 2 3;
    4 5 6;
    7 8 9;
];
disp("Testing partial pivoting with the matrix:")
disp(B)
myPartialPivot(B)

%% Question 3

% 1(a)
function U = mySymmetricPivot(A)
    [m, n] = size(A);
    if n ~= m
        error("`A` must be an n x n matrix, got: %d x %d", m, n)
    end

    for j = 1:n  % for each column

        % Locate the row with biggest absolute value.
```

1/1

1/1

1/1

Good...

```
        maxAbsValue = 0;
        maxRowIndex = 0;
        for i = j:m  % for each row from pivot
            absValue = abs(A(i,j));
            if absValue > maxAbsValue
                maxAbsValue = absValue;
                maxRowIndex = i;
            end
        end

        % If everything is zero below, skip.
        if maxAbsValue == 0
            continue
        end

        % Perform swap, if necessary.
        if maxRowIndex ~= j
            pivotRow = A(j,:);
            A(j,:) = A(maxRowIndex, :);
            A(maxRowIndex, :) = pivotRow;
            % pivotCol = A(:,j);
            % A(:,j) = A(:, maxRowIndex);
            % A(:, maxRowIndex) = pivotCol;
        end

        % Eliminate rows.
        % for i = j+1:m
        %     ratio = A(i,j) / A(j,j);
        %     A(i,:) = A(i,:) - ratio * A(j,:);
        % end
    end

    U = A;
end

A = [1 7 3; 7 4 5; 3 5 2]
mySymmetricPivot(A)
```