```
function X = gauss_elimination(A, B, m, n)
    operations = 0;
    X = zeros(n,1); % Initialize X

    % Sorting initially
    [A, B] = sort(A, B, 1, 1, m, n);

    for c = 1:n
        % Sorting A and B (max. diagonal element)
        [A, B] = sort(A, B, c, c, m, n);
        for r = m:-1:c+1
            if (A(r,c) ~= 0)
                factor = A(r, c)/A(c, c);  % 1 operation
                A(r,:) = A(r, :) - (factor*A(c, :));  % 2*n operations
                B(r) = B(r) - (factor*B(c));  % 2 operations
                operations = operations + (2*n) + 3;
            end
        end
    end

    % Back-Substitution
    X(n) = (B(n)/A(n,n));
    operations = operations + 1;
    for i = m-1:-1:1
        sum = 0;
        for j = n:-1:i+1
            sum = sum + (X(j)*A(i, j));
            operations = operations + 2;
        end
        X(i) = (B(i) - sum)/A(i, i);
        operations = operations + 2;
    end
    % printing number of operations
    disp('Number of Operations =')
    disp(operations)

end

% Sorting Function
function [mat1, mat2] = sort(A, B, rs, cs, m, n)
    for s = rs:m-1
        for r = rs:m-1
            if abs(A(r, cs)) < abs(A(r+1, cs))
                temp1 = A(r, :);
                A(r, :) = A(r+1, :);
                A(r+1, :) = temp1;
                temp2 = B(r);
                B(r) = B(r+1);
                B(r+1) = temp2;
            end
        end
    end
    mat1 = A;
    mat2 = B;
end
```