Computer Exercise 2.1.3

The following program will use the *naive_gauss* algorithm to solve $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A}$ is an $n \times n$ matrix defined by $[\mathbf{A}]_{ij} = i + j$ and $\mathbf{b}$ is an $n \times 1$ column vector defined by $b_i = i + 1$.

```
%seed for random integer
format default
rng('default')
s = rng;
%iniate parameters
n=randi([3, 6]);
A = zeros(n, n);
b = zeros(n, 1);

%satisfy conditions A(i, j) = i+j
% and b(i) = i + 1
for i = 1:n
    b(i) = i + 1;
    for j = 1:n
        A(i, j) = i + j;
    end
end
%run naive gauss algorithm
x = naive_gauss(A, b);
x
```

```
x = 6×1
   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
```

We see that $\mathbf{x}$ yields an infinite amount of solutions. This is to be expected since $det(\mathbf{A}) = 0$ as can be checked by the following command:

```
det(A)
```

```
ans = 0
```

```
function x = naive_gauss(A, b)
    n = length(b);
%forward elimination
    for k = 1:(n-1)
        for i = (k+1):n
            xmult = (A(i, k))/(A(k, k));
            A(i, k) = 0;
            for j = (k+1):n
                A(i, j) = A(i, j) - xmult*A(k, j);
                if abs(A(i, j)) < 10^(-12)
```

```
                    A(i, j) = 0;
                end
            end
            b(i) = b(i) - xmult*b(k);
        end
    end
%backwards substitution
    x = zeros(n, 1);
    x(n) = (b(n))/(A(n,n));
    for u = (n-1):-1:1
        sum = 0;
        for v = (u+1):n
            sum = sum + (A(u, v)*x(v));
        end
        x(u) = (b(u) - sum)/(A(u, u));
    end
end
```