```
% Consider the set of vectors in R5: B1 = {w1, w2, w3, w4, w5, w6} where
w1 = [-1; 1; 2; 4; 1];
w2 = [-1; 1; 2; 1; 1];
w3 = [3; 1; -1; 2; 0];
w4 = [2; 1; 0; 3; -1];
w5 = [5; 4; 1; 11; 3];
w6 = [1; 0; -1; 2; 1];

% Show B1 is a linearly dependent set.
% The set of vectors in R5: B1 is linearly dependent if there is
% a non-trivial solution to c1w1 + c2w2 + c3w3 + c4w4 + c5w5 + c6w6 = 0

% Create a system of equations using the vectors in B1
B1 = [w1(:), w2(:), w3(:), w4(:), w5(:), w6(:)];
zeroVectorR5 = [0; 0; 0; 0; 0];

% Create an augmented matrix wih the zero vector
B1Aug = cat(2, B1, zeroVectorR5);
% Row reduce B1Aug into RREF
B1RREF = rref(B1Aug)

% The system of equations is not linearly dependent as there is a non-zero
% solution to the system. The columns without leading 1's are linearly
% dependent
sprintf('As seen in the RREF of B1RREF, the system is consistent with a non-
zero solution');
sprintf('There is no leading variables for x5 and x6, they are free
 variables');

% Find a maximal linearly independent set B#1 of vectors from B1
sprintf('The maximal linearly independent set B''1 of vectors is {w1, w2, w3,
 w4}')

% For testing purposes only... making sure that there is a unique solution
B1_maximal = [w1(:), w2(:), w3(:), w4(:)];
B1_maximal_rref = rref(B1_maximal);

% Show that vectors from B1 that are NOT in B#1 set are contained in the span
 of B#1 (and hence, that span B1 = span B#1).
% I am using consistency in w5 = {w1, w2, w3, w4} and w6 = {w1, w2, w3, w4}
B1_w5_Aug = cat(2, [w1(:), w2(:), w3(:), w4(:)], w5)
B1_w5_Rref = rref(B1_w5_Aug)

B1_w6_Aug = cat(2, [w1(:), w2(:), w3(:), w4(:)], w6)
B1_w6_Aug_Rref = rref(B1_w6_Aug)
% Clearly consistent

% What is the dimension of span B1?
sprintf(['The dimension of the span is equal to how many independent vectors
 there are in Span B''1. In this case' ...
    'there are obviously 4 independent vectors. Thus the dimension is 4.'])
```

```
% Consider the set: B2 = {z1, z2, z3, z4, z5} where
z1 = [5; 2; 1; 7; 1];
z2 = [2; -1; 0; 0; 1];
z3 = [1; 2; 1; 1; 0];
z4 = [2; -4; -2; 4; 1];
z5 = [0; 1; 2; 3; -1];

% Create a system of equations using the vectors in B2
B2 = [z1(:), z2(:), z3(:), z4(:), z5(:)];

% Create an augmented matrix wih the zero vector
B2Aug = cat(2, B2, zeroVectorR5);

% Row reduce B1Aug into RREF
B2RREF = rref(B2Aug)
sprintf('As seen in B2RREF, the system is consistent with a non-zero
 solution');
sprintf('There is no leading variables for x4, it is a free variable');

% Find a maximal linearly independent set B#2 of vectors from B2
sprintf('The maximal linearly independent set B''1 of vectors is {z1, z2, z3,
 z5}')

% For testing purposes only... making sure that there is a unique solution
B2_maximal = [z1(:), z2(:), z3(:), z5(:)];
B2_maximal_rref = rref(B2_maximal);

% Find all the vectors in the intersection span B1 INTERSECTION span B2.
% Find the maximal linearly independent set for both!
% To find the intersection of the span, find the vectors in both... Set the
% linearly combination of both spans equal to each other.
B1B2_Maximal_System = [w1(:), w2(:), w3(:) w4(:), -z1(:), -z2(:), -z3(:), -
z5(:)];
B1B2_Aug = cat(2, B1B2_Maximal_System, zeroVectorR5);
B1B2_Rref = rref(B1B2_Aug);
B1B2_Rref = rats(B1B2_Rref)


% Show that this is a subspace of R5.
% By definition, any span is a subspace. In addition, the intersection of
% two spans will also be a subspace. However, to be more verbose, a
% subspace is non-empty, closed under addition. and closed under scalar
% multiplcation. It obviously contains the non-zero vector. (Say for marks)

% For

% For closed under addition, if we let w1 and w2 be some vectors in B1 and
% B2, then that means that

%
% Show w1 + w2 is in the span of B1 and B2. if w1 and w2 are
% in span(B2). That implies that both w1 and w2 are in the intereciton of
% B2. that implies that w1 and w2 are in the intersection.
% Similarly for scalar multiplication
```

```
% 1. Part 1 - Solutions of systems of linear equations
    % By considering random matrices of appropriate sizes, find "emperical
 evidence" that
    % substantiate the following statements. (In this part, m and n are both
 integers and both are greater than 6, so 7 or larger).
    % Give examples of exceptions for each case (but for this part, you can
 use integers m and n that need only be larger than 2, so 3 or larger).

    % (a) A system of n linear equations in n unknowns typically has a unique
 solution.
    % (b) A system of m linear equations in n unknowns, where m > n, typically
 has no solution.
    % (c) A system of m linear equations in n unknowns, where m < n, typically
 has many solutions.

% (a) Generate 100 random matrices with integers of size n x n where n is
 between 8 and 20.
    % Check if a unique solution exists for each system of linear equations Ab

    sprintf('NOW STARTING PART A HERE ------------------------------------')
    % Use rank to determine consistency in this case: Theorem 1.1 states
    % that a linear system is consistent IFF the rank of the coefficient
    % matrix is equal to the rank of the augmented matrix
    count = 0;
    for i = 1:100
        % Make a selection of n, where 8 <= n <= 20
        n = randi([8, 20]);
        % Generate a matrix of size nxn with real numbers
        A = rand(n);
        % Generate a vector of size n*1 with real numbers to represent the
 right most column
        b = rand(n, 1);
        % Generate our augmented matrix by concatenating matrix A and vector b
        Ab = cat(2, A, b);

        % Verification for testing purposes only... remove after
        Ab_rref = rref(Ab);

        % If rank(Coeffient matrix == augmented matrix), we have a unique
        % solution because the system is consistent and there are no free
 variables
        if rank(A) == rank(Ab)
            count = count + 1;
        end
    end

    % Output the final count of how many randomly generated equations have
 unique solutions
    sprintf('Out of 100 randomly generated systems of linear equations of size
 m=n, %d had a unique solution', count)
```

```
    % Print out MY generated exception to the rule
    sprintf('Example of a system of linear equations (augmented matrix) that
 does not have a unique solution when m = n')
    A = [1 2 3; 1 2 3; 1 2 3];
    b = [1; 2; 3];
    Ab = cat(2, A, b);
    rref(Ab)

    % If the rank of the coefficient matrix is LESS than the rank of the
 augmented matrix, there are no solutions
    if (rank(A) < rank(Ab))
        sprintf('Exception: This m=n system is inconsistent and therefore has
 no solution')
    end

% (b) Case 2: m > n typically has no solution
    % Generate 100 random matrices with integers of size m x n where m is
 between 8 and 20 and n is between 8 and 20
    % Check if a unique solution exists for each system of linear equations
    sprintf('NOW STARTING PART B HERE --------------------------------')
    % A system of equations is INCOSISTENT or has no solutions when the
    % rank of the coefficient matrix is less than the rank of the augmented
    % matrix
    count2 = 0;
    for i = 1:100
        % Make a selection for m, where 8 <= m <= 20
        m = randi([15, 20]);
        % Make a selection for n, where m > n and greater than 7 (aka 7-15)
        n = randi([7, m-1]);

        % Generate a matrix of size mxn with real numbers
        % Check documentation here to see which parameter matches to m or n
        B = rand(m, n);
        % Generate a vector of size n with real numbers to represent the
        % right most column
        c = rand(m, 1);
        % Generate our augmented matrix by concatenating matrix B and
        % vector b
        Bc = cat(2, B, c);

         % Verification for testing purposes only... remove after
        Bc_rref = rref(Bc);

        % If the rank of the coefficient matrix is LESS than the rank of
        % the augmented matrix, there are no solutions
        if rank(B) < rank(Bc)
            count2 = count2 + 1;
        end
    end
    sprintf('Out of 100 randomly generated systems of linear equations of size
 m > n, %d did not have a unique solution', count2)

    % Check these examples again later
```

```
   sprintf('Example of a system of linear equations (augmented matrix) does
 not follow the rule that m > n typically has no solution')
   m2 = 4;
   n2 = 3;
   B2 = [1 2 3; 4 5 6; 7 8 9; 10 11 12];
   c2 = [1; 2; 3; 4];
   Bc2 = cat(2, B2, c2)
   Bc_rref2 = rref(Bc2)

   % This is an example of a system of linear equations that does not follow
 the rule that m > n typically has no solution
   sprintf('As seen in the RREF of Bc2, the system is consistent')

% (c) Case 3: m < n typically has many solutions
   sprintf('NOW STARTING PART C HERE ---------------------------------')
   count3 = 0;
   for i = 1:100
       % Generate an integer for n from [12 to 20]
       n = randi([12, 20]);
       % Generate an integer from [7 to n-1]
       m = randi([7, n-1]);

       % Generate a matrix of size mxn with real numbers
       C = rand(m, n);

       % Generate a vector of size m x 1 with real numbers
       D = rand(m, 1);

       % Concentrate matrix C and vector Cd to create an augmented matrix
       CD = cat(2, C, D);

       % If the rank of the coefficient matrix is LESS than the rank of
       % the augmented matrix, there are no solutions
       % BREAK
       if rank(C) < rank(CD)
           break
       end

       Cd_rref = rref(CD);
       % Since we know the system of equations is consistent, let's check if
 the solution has many solutions
       % If # of variables (columns) - rank is > 0, then there are
       % infinitely many solutions....
       if n - rank(Cd_rref) > 0
           count3 = count3 + 1;
       end
   end
   sprintf('Out of 100 randomly generated systems of linear equations of size
 m < n, %d had many solutions', count3)

   sprintf('Example of a system of linear equations (augmented matrix) that
 size m < n that is inconsistent and therefore has no solution')
   m3 = 3;
   n3 = 4;
```

```
    C3 = [1 2 3 4; 5 6 7 8; 9 10 11 12];
    D3 = [1; 2; 4];
    CD3 = cat(2, C3, D3)
    Cd_rref3 = rref(CD3)


B1RREF =

     1     0     0     0     2     1     0
     0     1     0     0     0    -1     0
     0     0     1     0     3     1     0
     0     0     0     1    -1    -1     0
     0     0     0     0     0     0     0


ans =

    'The maximal linearly independent set B'1 of vectors is {w1, w2, w3, w4}'


B1_w5_Aug =

    -1    -1     3     2     5
     1     1     1     1     4
     2     2    -1     0     1
     4     1     2     3    11
     1     1     0    -1     3


B1_w5_Rref =

     1     0     0     0     2
     0     1     0     0     0
     0     0     1     0     3
     0     0     0     1    -1
     0     0     0     0     0


B1_w6_Aug =

    -1    -1     3     2     1
     1     1     1     1     0
     2     2    -1     0    -1
     4     1     2     3     2
     1     1     0    -1     1


B1_w6_Aug_Rref =

     1     0     0     0     1
     0     1     0     0    -1
     0     0     1     0     1
     0     0     0     1    -1
     0     0     0     0     0
```

```
ans =

    'The dimension of the span is equal to how many independent vectors there
 are in Span B'1. In this casethere are obviously 4 independent vectors. Thus
 the dimension is 4.'


B2RREF =

     1     0     0     1     0     0
     0     1     0     0     0     0
     0     0     1    -3     0     0
     0     0     0     0     1     0
     0     0     0     0     0     0


ans =

    'The maximal linearly independent set B'1 of vectors is {z1, z2, z3, z5}'


B1B2_Rref =

  5×126 char array

    '       1             0             0             0             0
   7/9           4/9           2/9           0      '
    '       0             1             0             0             0
   5/9        -29/18          -5/9           0      '
    '       0             0             1             0             0
   4/3        -11/12           7/6           0      '
    '       0             0             0             1             0
    1           -3/4          -3/2           0      '
    '       0             0             0             0             1
   4/3         -5/12           1/6           0      '


ans =

    'NOW STARTING PART A HERE -------------------------------------'


ans =

    'Out of 100 randomly generated systems of linear equations of size m=n,
 100 had a unique solution'


ans =

    'Example of a system of linear equations (augmented matrix) that does not
 have a unique solution when m = n'
```

```
ans =

     1     2     3     0
     0     0     0     1
     0     0     0     0


ans =

    'Exception: This m=n system is inconsistent and therefore has no solution'


ans =

    'NOW STARTING PART B HERE ----------------------------------'


ans =

    'Out of 100 randomly generated systems of linear equations of size m > n,
  100 did not have a unique solution'


ans =

    'Example of a system of linear equations (augmented matrix) does not
  follow the rule that m > n typically has no solution'


Bc2 =

     1     2     3     1
     4     5     6     2
     7     8     9     3
    10    11    12     4


Bc_rref2 =

    1.0000         0   -1.0000   -0.3333
         0    1.0000    2.0000    0.6667
         0         0         0         0
         0         0         0         0


ans =

    'As seen in the RREF of Bc2, the system is consistent'


ans =
```

```
    'NOW STARTING PART C HERE ---------------------------------'


ans =

    'Out of 100 randomly generated systems of linear equations of size m < n,
 100 had many solutions'


ans =

    'Example of a system of linear equations (augmented matrix) that size m <
 n that is inconsistent and therefore has no solution'


CD3 =

     1     2     3     4     1
     5     6     7     8     2
     9    10    11    12     4


Cd_rref3 =

     1     0    -1    -2     0
     0     1     2     3     0
     0     0     0     0     1
```

Published with MATLAB® R2022a