Table of Contents


```
%%%% MATLAB PROJECT 1, MATH461, LINEAR ALGEBRA FOR SCIENTIST AND ENGINEERS
%%%% Romeo Perlstein, 2/20/2023
%%%% UID: 118030685, section 0123
```

Question 0

no, out submission needs to inlcude both commands AND output

Question 1

a)

```
A = [2 2 1 9 ; 4 10 1 24 ; 10 4 -3 -15]
```

b)

reduce the first column

```
A(2,:) = -2*A(1,:) + A(2,:)
A(3,:) = -5*A(1,:) + A(3,:)

% reduce the second column
A(3,:) = A(3,:) + A(2,:) % REF achieved!
```

c)

reduce to RREF now

```
A(1,:) = -2/6*A(2,:) + A(1,:)
A(2,:) = -1/9*A(3,:) + A(2,:)
A(1,:) = (4/(3*9))*A(3,:) + A(1,:)
% now make dem der pivots 1
A(1,:) = 1/2*A(1,:)
A(2,:) = 1/6*A(2,:)
A(3,:) = -1/9*A(3,:) %RREF achieved!
```

d)

```
%%%%% solutions
% q1_x1 = -1/2
% q1_x2 = 2
% q1_x3 = 6
```

```
A =

     2     2     1     9
     4    10     1    24
    10     4    -3   -15


A =

     2     2     1     9
     0     6    -1     6
    10     4    -3   -15


A =

     2     2     1     9
     0     6    -1     6
     0    -6    -8   -60


A =
```

```
     2     2     1     9
     0     6    -1     6
     0     0    -9   -54


A =

    2.0000         0    1.3333    7.0000
         0    6.0000   -1.0000    6.0000
         0         0   -9.0000  -54.0000


A =

    2.0000         0    1.3333    7.0000
         0    6.0000         0   12.0000
         0         0   -9.0000  -54.0000


A =

     2     0     0    -1
     0     6     0    12
     0     0    -9   -54


A =

    1.0000         0         0   -0.5000
         0    6.0000         0   12.0000
         0         0   -9.0000  -54.0000


A =

    1.0000         0         0   -0.5000
         0    1.0000         0    2.0000
         0         0   -9.0000  -54.0000


A =

    1.0000         0         0   -0.5000
         0    1.0000         0    2.0000
         0         0    1.0000    6.0000
```

Question 2

a)

```
B = [1 -6 0 -2 3 7 ; -4 27 -9 4 2 -28 ; 1 -6 0 2 -5 13 ; 2 -3 -27 -24 64 2]
```

```
B_OG = B
```

b)

reduce to row echelon form (REF)

```
B(2,:) = 4*B(1,:) + B(2,:)
B(3,:) = -1*B(1,:) + B(3,:)
B(4,:) = -2*B(1,:) + B(4,:) % column 1 obliterated

B(4,:) = -3*B(2,:) + B(4,:) % column 2 & 3 obliterated

B(4,:) = 2*B(3,:) + B(4,:) % REF achieved
```

c)

reduce to reduce row echelon form (RREF), because I have to (I WANT to)

```
B(1,:) = 2*B(2,:) + B(1,:)
B(1,:) = (10/4)*B(3,:) + B(1,:)
B(2,:) = B(3,:) + B(2,:)

% now make da pivots 1
B(2,:) = (1/3)*B(2,:)
B(3,:) = (1/4)*B(3,:)
```

d)

use the rref command to check that my answer is correct

```
matrix_ans = rref(B_OG)
if (matrix_ans == B)
    fprintf("we are all good \n")
end
```

e)

```
syms q2_x3 q2_x5

% v = [22, 2, 0, 1.5, 0] + q2_x3*[18, 3, 1, 0, 0] + q2_x5*[-11, -2, 0, 2, 1]
```

```
B =

     1    -6     0    -2     3     7
    -4    27    -9     4     2   -28
     1    -6     0     2    -5    13
     2    -3   -27   -24    64     2


B_OG =
```

```
     1    -6     0    -2     3     7
    -4    27    -9     4     2   -28
     1    -6     0     2    -5    13
     2    -3   -27   -24    64     2


B =

     1    -6     0    -2     3     7
     0     3    -9    -4    14     0
     1    -6     0     2    -5    13
     2    -3   -27   -24    64     2


B =

     1    -6     0    -2     3     7
     0     3    -9    -4    14     0
     0     0     0     4    -8     6
     2    -3   -27   -24    64     2


B =

     1    -6     0    -2     3     7
     0     3    -9    -4    14     0
     0     0     0     4    -8     6
     0     9   -27   -20    58   -12


B =

     1    -6     0    -2     3     7
     0     3    -9    -4    14     0
     0     0     0     4    -8     6
     0     0     0    -8    16   -12


B =

     1    -6     0    -2     3     7
     0     3    -9    -4    14     0
     0     0     0     4    -8     6
     0     0     0     0     0     0


B =

     1     0   -18   -10    31     7
     0     3    -9    -4    14     0
     0     0     0     4    -8     6
     0     0     0     0     0     0
```

```
B =

     1     0   -18     0    11    22
     0     3    -9    -4    14     0
     0     0     0     4    -8     6
     0     0     0     0     0     0


B =

     1     0   -18     0    11    22
     0     3    -9     0     6     6
     0     0     0     4    -8     6
     0     0     0     0     0     0


B =

     1     0   -18     0    11    22
     0     1    -3     0     2     2
     0     0     0     4    -8     6
     0     0     0     0     0     0


B =

    1.0000         0  -18.0000         0   11.0000   22.0000
         0    1.0000   -3.0000         0    2.0000    2.0000
         0         0         0    1.0000   -2.0000    1.5000
         0         0         0         0         0         0


matrix_ans =

    1.0000         0  -18.0000         0   11.0000   22.0000
         0    1.0000   -3.0000         0    2.0000    2.0000
         0         0         0    1.0000   -2.0000    1.5000
         0         0         0         0         0         0

we are all good
```

Question 3

```
format short
A = [3 -6 2 6 ; -5 -4 4 4 ; 3 2 3 6]
```

a)

use rref() to reduce the matrix

```
A_ans = rref(A)
```

b)

```
q3_x1 = A_ans(1,4)
q3_x2 = A_ans(2,4)
q3_x3 = A_ans(3,4)
% x1 = .5872
% x2 = -.1927
% x3 = 1.5413
```

c)

```
format rat
A_ans_frac = A_ans
```

d)

```
q3_x1 = A_ans_frac(1,4)
q3_x2 = A_ans_frac(2,4)
q3_x3 = A_ans_frac(3,4)
% x1 = 64/109
% x2 = -21/109
% x3 = 168/109
```

```
A =

     3    -6     2     6
    -5    -4     4     4
     3     2     3     6


A_ans =

    1.0000         0         0    0.5872
         0    1.0000         0   -0.1927
         0         0    1.0000    1.5413


q3_x1 =

    0.5872


q3_x2 =

   -0.1927


q3_x3 =

    1.5413
```

```
A_ans_frac =

     1            0            0             64/109
     0            1            0            -21/109
     0            0            1            168/109


q3_x1 =

    64/109


q3_x2 =

   -21/109


q3_x3 =

   168/109
```

Question 4

```
format short
syms x1 x2 x3 x4
270*x1 + 51*x2 + 70*x3 +260*x4 == 400
10*x1 + 5.4*x2 + 15*x3 +9*x4 == 30
2*x1 + 5.2*x2 + 0*x3 + 5*x4 == 10

% first, set up matrix (ez pz)
% wants 400 calories, 30g protein, 10g fiber, so
Mac_n_matrix = [270 51 70 260 400; 10 5.4 15 9 30; 2 5.2 0 5 10]
```

a)

RREF the sucker do see what each coefficient should go for each place

```
Mac_n_ans = rref(Mac_n_matrix)

% she should eat
syms q4_x4
format short %% format short not working???
q4_x = [Mac_n_ans(1,5), Mac_n_ans(2,5), Mac_n_ans(3,5)] + q4_x4*[-
(Mac_n_ans(1,4)), -(Mac_n_ans(2,4)), Mac_n_ans(3,4)]
```

b)

I'm either blind or the textbook did not list the specs of the whole wheat mac and cheese, so I looked up the nutrition facts

```
270*x1 + 51*x2 + 70*x3 +280*x4 == 400
```

```
10*x1 + 5.4*x2 + 15*x3 +10*x4 == 30
2*x1 + 5.2*x2 + 0*x3 + 4*x4 == 1
% following same procedure
Mac_n_matrix_whole = [270 51 70 280 400; 10 5.4 15 10 30; 2 5.2 0 4 10]
Mac_n_ans_whole = rref(Mac_n_matrix_whole)
q4_x = [Mac_n_ans_whole(1,5), Mac_n_ans_whole(2,5), Mac_n_ans_whole(3,5)]
 + q4_x4*[-(Mac_n_ans_whole(1,4)), -(Mac_n_ans_whole(2,4)),
 Mac_n_ans_whole(3,4)]


ans =

270*x1 + 51*x2 + 70*x3 + 260*x4 == 400


ans =

10*x1 + (27*x2)/5 + 15*x3 + 9*x4 == 30


ans =

2*x1 + (26*x2)/5 + 5*x4 == 10


Mac_n_matrix =

  270.0000   51.0000   70.0000  260.0000  400.0000
   10.0000    5.4000   15.0000    9.0000   30.0000
    2.0000    5.2000         0    5.0000   10.0000


Mac_n_ans =

    1.0000         0         0    0.9053    0.9858
         0    1.0000         0    0.6134    1.5439
         0         0    1.0000   -0.2243    0.7870


q4_x =

[139/141 - (86*q4_x4)/95, 457/296 - (303*q4_x4)/494, 133/169 -
 (107*q4_x4)/477]


ans =

270*x1 + 51*x2 + 70*x3 + 280*x4 == 400


ans =

10*x1 + (27*x2)/5 + 15*x3 + 10*x4 == 30
```

```
ans =

2*x1 + (26*x2)/5 + 4*x4 == 1


Mac_n_matrix_whole =

  270.0000   51.0000   70.0000  280.0000  400.0000
   10.0000    5.4000   15.0000   10.0000   30.0000
    2.0000    5.2000         0    4.0000   10.0000


Mac_n_ans_whole =

    1.0000         0         0    1.0004    0.9858
         0    1.0000         0    0.3845    1.5439
         0         0    1.0000   -0.1387    0.7870


q4_x =

[139/141 - (2775*q4_x4)/2774, 457/296 - (213*q4_x4)/554, 133/169 -
 (33*q4_x4)/238]
```

Question 5

a)

constructing the appropriate matrix and RREF'ing it

```
Matrix_ = [13.2 4.8 14.2 ; .9  -2.4  -.1 ; -2.9 -5.6 -4.9]
Matrix_q5_RREF = rref(Matrix_)
```

b)

The vector [14.2, -.1, -4.9] (lets call it w) is in the span of the two given vectors because it's last row does not have a pivot, and more importantly, the solution to the matrix is CONSISTENT, meaning there are given coefficients that make w a linear combinaiton of one of the vectors in the span

c)

```
Matrix_dependence_check = [13.2 4.8 14.2 0 ; .9  -2.4  -.1 0 ; -2.9 -5.6 -4.9
 0]
Matrix_dep_chk_sol = rref(Matrix_dependence_check)
% The solution to the matrix is linearly dependent, because its solution,
% when setting each row equal to 0, meaning it has more than just the
% trivial solution of the zero vector
```

```
Matrix_ =

   13.2000    4.8000   14.2000
    0.9000   -2.4000   -0.1000
   -2.9000   -5.6000   -4.9000


Matrix_q5_RREF =

    1.0000         0    0.9333
         0    1.0000    0.3917
         0         0         0


Matrix_dependence_check =

   13.2000    4.8000   14.2000         0
    0.9000   -2.4000   -0.1000         0
   -2.9000   -5.6000   -4.9000         0


Matrix_dep_chk_sol =

    1.0000         0    0.9333         0
         0    1.0000    0.3917         0
         0         0         0         0
```

Question 6

```
format short
```

a)

```
syms a b
```

b)

```
Matrix_for_use = [6 2 a ; -5 -9 b]
matrix_for_use_sol = rref(Matrix_for_use)
```

c)

```
w1 = matrix_for_use_sol(1,3)
w2 = matrix_for_use_sol(2,3)
% w1 = 9a/44 + 3b/22
% w2 = -5a/44 - 3b/22
```

```
Matrix_for_use =

[ 6,  2, a]
```

```
[-5, -9, b]


matrix_for_use_sol =

[1, 0,        (9*a)/44 + b/22]
[0, 1, - (5*a)/44 - (3*b)/22]


w1 =

(9*a)/44 + b/22


w2 =

- (5*a)/44 - (3*b)/22
```

Question 7

finding linear dependence (and self-independenc... not really)

a)

```
Matrix_A = [-1 -5 -4 5 2 0 ; 0 -2 2 3 -7 0 ; -7 2 -4 10 8 0 ; 2 -5 6 4 7 0]
Matrix_A_ans = rref(Matrix_A)
```

b)

set x4 as any real number and hopefully get 0

```
%x1 = 2x4
%x2 = x4
%x3 = -1/2x4
%x5 = 0
%x4 = free
q7_x4 = 2 % went through and tested multiple numbers too!
x = q7_x4*[2, 1, -.5, 1, 0]
result = x(1)*[-1,0,-7,2] + x(2)*[-5,-2,2,-5] + x(3)*[-4,2,-4,6] +
 x(4)*[5,3,10,4] + x(5)*[2,-7,8,7]
if( result == 0)
    fprintf("we are DONE baybee (like austin powers) \n")
else
    fprintf("its so over (bad way) \n")
end
```

d)

One theorem from 1.7 states that if p>n, as in the number of vectors is greated than the dimension each vector is in (think R^n), then ANY set of vectors in R^n is linearly dependent. From observation, the matrix formed out of the given vectors clearly has a p value greater than n (to be precise, p = 5, n = 4), so it must be linearly dependent

e)

checking for span

```
syms b1 b2 b3 b4
Matrix_span_check = [-1 -5 -4 5 2 b1 ; 0 -2 2 3 -7 b2 ; -7 2 -4 10 8 b3 ; 2 -5
 6 4 7 b4]
Matrix_span_sol = rref(Matrix_span_check)
% The vectors do span R^4 because the system is consistent, as in there is
% at least more than one solution to the system (in this case infinite
% since there is a free variable)
```

```
Matrix_A =

    -1    -5    -4     5     2     0
     0    -2     2     3    -7     0
    -7     2    -4    10     8     0
     2    -5     6     4     7     0


Matrix_A_ans =

    1.0000         0         0   -2.0000         0         0
         0    1.0000         0   -1.0000         0         0
         0         0    1.0000    0.5000         0         0
         0         0         0         0    1.0000         0


q7_x4 =

     2


x =

     4     2    -1     2     0


result =

     0     0     0     0

we are DONE baybee (like austin powers)

Matrix_span_check =

[-1, -5, -4,  5,  2, b1]
[ 0, -2,  2,  3, -7, b2]
[-7,  2, -4, 10,  8, b3]
[ 2, -5,  6,  4,  7, b4]
```

```
Matrix_span_sol =

[1, 0, 0,  -2, 0,           (5*b1)/157 - (29*b2)/157 - (24*b3)/157 -
 (3*b4)/157]
[0, 1, 0,  -1, 0,          (2*b3)/785 - (63*b2)/785 - (92*b1)/785 -
 (39*b4)/785]
[0, 0, 1, 1/2, 0, (317*b2)/3140 - (347*b1)/3140 + (127*b3)/3140 +
 (271*b4)/3140]
[0, 0, 0,   0, 1,     (3*b1)/1570 - (143*b2)/1570 + (17*b3)/1570 +
 (61*b4)/1570]
```

Published with MATLAB® R2022b