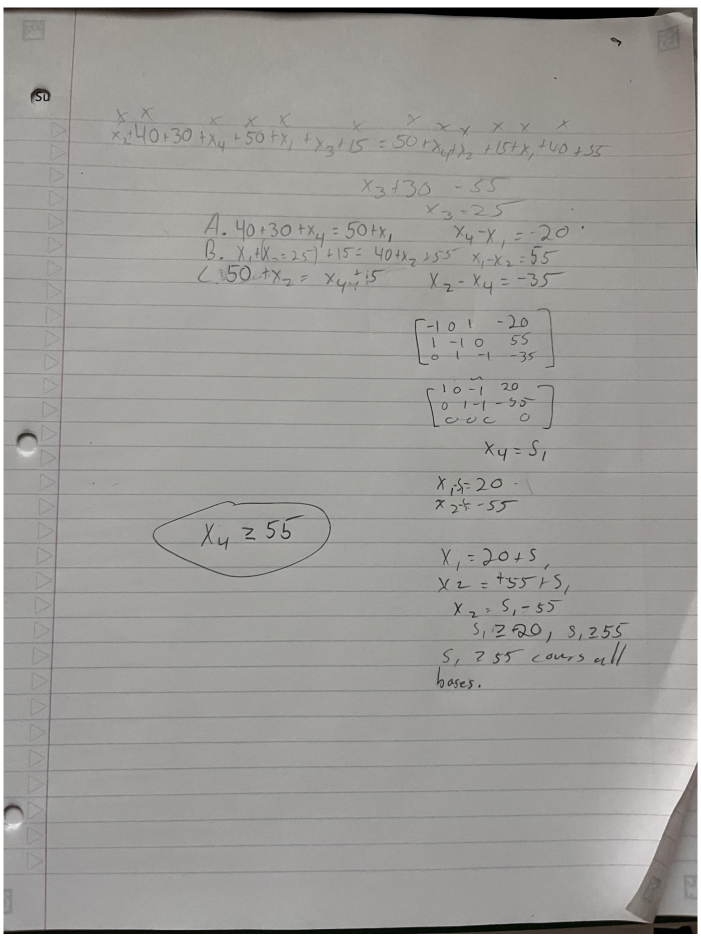
Orion Gregory
This is my work and signed my by me, OG (imagine thats my signature)
1.

This problem required me to set up a system of equations involving all of the vectors that flow in and out of each point, A B and C. Thinking of traffic as a flow from each point allows us to create a system of equations that we can solve using matrices. If we set the total flow of each vector going in and out of each point all together in one system, IE.

40 + 30 + x4 + 50 + x1 + x2 + 15 + x3 = 50 + x4 + x1 + x2 + 15 + 40 + 55

This allows us to find x3, which equals 25.

Once we have found X3 we can create three equations, with three variables. When we put each equation into reduced echelon form, we find that x4 is our free variable, and that x1 = 20 + S1, (S1 being the parameter for x4), and x2 = S1 - 55. In order for the entire system to work, the amount of flow into a point must equal the amount of flow exiting the point, and with this in mind, s1 must be a large enough value to satisfy this principle. Solving for the greatest S1 allows us to find the minimum flow between C and A because S1 is the parameter for x4, and x4 is defined by the flow in between C and A. We find that S1 must equal 55 or be greater than 55 from the equation X2 = S1 - 55.



$x_2 + 40 + 30 + x_4 + 50 + x_1 + x_3 + 15 = 50 + x_4 + x_2 + 15 + x_1 + 40 + 55$

$x_3 + 30 = 55$

$x_3 = 25$

A. $40 + 30 + x_4 = 50 + x_1$ $\quad x_4 - x_1 = -20$

B. $x_1 + (x_3 = 25) + 15 = 40 + x_2 + 55$ $\quad x_1 - x_2 = 55$

C. $50 + x_2 = x_4 + 15$ $\quad x_2 - x_4 = -35$

$$\begin{bmatrix} -1 & 0 & 1 & -20 \\ 1 & -1 & 0 & 55 \\ 0 & 1 & -1 & -35 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -1 & 20 \\ 0 & 1 & -1 & -55 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$x_4 = S_1$

$x_1 - S_1 = 20$

$x_2 - S_1 = -55$

$x_4 \geq 55$

$x_1 = 20 + S_1$

$x_2 = +55 + S_1$

$x_2 = S_1 - 55$

$S_1 \geq 20, \; S_1 \geq 55$

$S_1 \geq 55$ covers all bases.

2.

I decided to encode Hello World. This was done by changing each character of the phrase into a numerical value, associated with its position in the alphabet. Then a matrix was dynamically created, with a dynamic amount of columns and a fixed amount of rows, corresponding to the amount of columns that was given by our encoding matrix A. Doing matrix multiplication of A and our message matrix gave us an encoded message! Granted it is in matrix form and is encoded so its information is useless without the context of the encoding matrix.

```
matrixComplete =

    30    69    71    47
    29    72   103    17
    18    42    53    20
```

2.a - 2.c

The same process as the Hello World message was done to encode the message ' I love linear algebra'.

```
encodedMessage =

   -90    10   -30   -86   -42  -107
   -99   103   -34  -119   -39   -79
   267   -15    94   247   110   321
  -345    52  -119  -332  -150  -401
```

```
>> This is I Love Linear Algebra!
```

Really is no different. Decoding of the two number arrays was done by finding the inverse of the B encoding matrix, which can be called the decoding matrix, and multiplying the two matrices together, it was important to make sure that each number array could be allocated correctly to be multiplied by the 4x4 b decoding matrix. This was done by allocating a dynamic matrix with dynamic rows to ensure that the multiplication between the encoded message and the decoder array would go through without error.

```
uncodedMessage =

    19    14     5    15    15     5    20
     5    27    19    14    14    19    23
    22    17    20    19    27    20    15
     5    21     9    27    20    27    27

>> uncodedMessage1

uncodedMessage1 =

     3     7     6    19    19    16    27     5
     8     5    27     9    21     1     3    27
     1    27     2    19     2     3     1    27
    14    15     1    27    19     5    19    27
```

C. Any matrix with the determinant value of zero is not invertible. Because of this, in order to have a useful encoding matrix, you have to be able to find its inverse to decode any message that you encoded with the encoding matrix. It would be impossible to decode any message encoded with a matrix that has a determinant of 0.

```
>> C = [6 2 0 1; 3 1 0 2; 0 3 2 0; 0 0 0 -7];
int = det(C);
>>
>> int

int =

     0
```