Activity: Creating a Secret Message

Encoding the Secret Message

You need to get a secret message to a friend and want to encode the message for privacy. One method is to map the letters of the alphabet to the numbers 0 to 25, and include a "space" as number 26:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Space
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

If we encode the word **APPLES** into the a numerical code we get the list of numbers { 0, 15, 15, 11, 4, 18}. To encode this sequence of numbers we can use matrix multiplication with an *encoding matrix*. To do this make the message numbers into a 2×3 matrix using the first two elements as the first column, etc. This will be the message matrix: messageM = $\begin{pmatrix} 0 & 15 & 4 \\ 15 & 11 & 18 \end{pmatrix}$.

In[1]:=
```
messageM = {{0, 15, 4}, {15, 11, 18}}
```

Out[1]=
```
{{0, 15, 4}, {15, 11, 18}}
```

To encode the message, create a 2×2 matrix as the encoding matrix, i.e, encodingM = $\begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}$.

In[2]:=
```
encodingM = {{2, 1}, {2, 3}}
```

Out[2]=
```
{{2, 1}, {2, 3}}
```

Next, encode the message by calculating **encodingM.messageM**

In[3]:=
```
encodingM.messageM
```

Out[3]=
```
{{15, 41, 26}, {45, 63, 62}}
```

Convert back from numerical values to letters for the encoded text. Problem!!! Some values are larger than 26.

Modulo Arithmetic

Modular arithmetic is a type of math in which the result is always in a finite set of natural numbers, that is, in **modulo 7** only the values {0, 1, 2, 3, 4, 5, 6} are the only results. If we calculate 2×3 we get 6 which is still in the set of modulo numbers. If we calculate 2×5 we get 10, which is not in the list. However, start counting at 1 up to 6, and then continue around with 7 at the 0 place, and continue until you reach 10. You should be at 3. This means $10 = 3 \pmod 7$ or $10 \overset{7}{=} 3$, or 10 mod 7 = 3. Also, $12 \times 2 \bmod 7 = 3$.

Another method to calculate modulo values is the remainder after long division of the value by the modulus.

Example 1 Find $5 \times 6 \pmod 8$.

Example 2 Find $\begin{pmatrix} 3 & 8 \\ -1 & 4 \end{pmatrix}\begin{pmatrix} 5 & 4 \\ 3 & -2 \end{pmatrix} \pmod{11}$

Using modulo arithmetic, we can convert our coded message values to modulo 27 and find the coded text:

```
In[4]:= codedM = Mod[encodingM.messageM, 27]
Out[4]= {{15, 14, 26}, {18, 9, 8}}
```

This gives us the characters: {15, 18, 14, 9, 26, 8}, or the text: **PSOJ I**. Now you can transmit your secret word.

Decoding a Message

Decoding the message is just the reverse of encoding: (1) convert the text to a numerical matrix (2) multiply by the **inverse** of the decoding matrix, (3) convert back to the original text message.

The coded text is **PSOJ I**, which converts to the matrix $M = \begin{pmatrix} 15 & 14 & 26 \\ 18 & 9 & 8 \end{pmatrix}$. Next, we need to find the decoding matrix. This is the inverse of the encoding matrix:

```
In[5]:= Inverse[encodingM]
Out[5]= {{3/4, -1/4}, {-1/2, 1/2}}
```

Here we have another problem: the inverse has fractions and negative values which may result in a decoded message with fractions and negatives. To fix this, we need to convert the fractions to modulo 27.

Converting Fractions to Modulo n

Suppose you need to find what the fraction $\frac{3}{4}$ is modulo 27, i.e, $\frac{3}{4} \overset{27}{\cong} m$. Multiplying each side by 4, we get $3 \overset{27}{\cong} 4m$. This means we need to find a value m such that $4m \overset{27}{\cong} 3$. This is not obvious, but listing out all of the numbers the equal 3 mod(27), gives {3, 30, 57, 84, 111, etc}. In this list we see $84 = 4m$ giving $m = 21$. Therefore, $\left(\frac{3}{4} = 21\right) \pmod{27}$ or $\frac{3}{4} \overset{27}{\cong} 21$.

Example 3 Find m such $\frac{2}{5} \pmod 7 = m$ and demonstrate this by multiplying $\frac{2}{5} \times 25 \pmod 7$ and $m \times 25 \pmod 7$.

We can also get *Mathematica* to find modulus fractions. To find $\frac{3}{4} = m \pmod{27}$, we solve $3 = 4m \pmod{27}$ as follows:

```
In[6]:= Solve[3 == 4 m, m, Modulus → 27]
Out[6]= {{m → 21}}
```

However, *Mathematica* can calculate the inverse matrix and convert to mod 27 in one command:

```
In[7]:= decipherM = Inverse[encodingM, Modulus → 27]
Out[7]= {{21, 20}, {13, 14}}
```

We can finally use this matrix to decipher our coded matrix, and convert to modulus 27:

```
In[8]:= Mod[decipherM.codedM, 27] // MatrixForm
Out[8]//MatrixForm=
( 0  15  4 )
( 15 11  18 )
```

Recovering the message text we get: {0, 15, 15, 11, 4, 18} = **APPLES**.

Your Mission

You have intercepted the encrypted message below:

CFAZGUJCCZESGNMRLUGUUZOVFAEC

From previously captured messages, you suspect that the last four characters in the message is the tell-tale sign **BOND**. You need to use this information to recover the original encoding matrix: encodingM = $\begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix}$. Set up and solve the equation $\begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix}\begin{pmatrix} B & N \\ O & D \end{pmatrix} = \begin{pmatrix} F & E \\ A & C \end{pmatrix}$ for e_1, e_2, e_3, and e_4 using **Modulus→27** (with each letter being the appropriate number). From here you can find the decoding matrix and decode the secret message.

Example: To solve: $\begin{pmatrix} e1 & e2 \\ e3 & e4 \end{pmatrix}\begin{pmatrix} 4 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 5 & 17 \\ 4 & 9 \end{pmatrix} \pmod{27}$

```
In[9]:= Solve[{{e1, e2}, {e3, e4}}.{{4, 2}, {3, 4}} == {{5, 17}, {4, 9}}, {e1, e2, e3, e4}, Modulus → 27]
Out[9]= {{e1 → 5, e2 → 22, e3 → 7, e4 → 19}}
```

Therefore, the encoding matrix used in this example is $\begin{pmatrix} 5 & 22 \\ 7 & 19 \end{pmatrix}$.

Use the previous commands and concepts to decode the secret message above. Find the inverse of the encoding matrix by hand (non modulo) as well as having *Mathematica* find the inverse modulo 27.