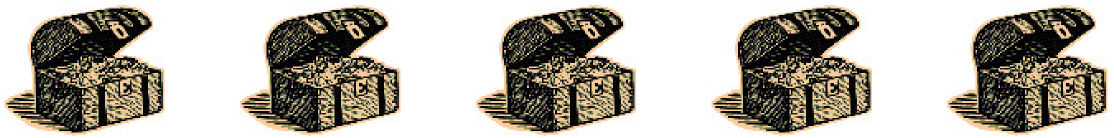

Introduction to University Mathematics Treasure Hunt

Now that the course is over, we can turn to lighter things. “What was this all for?” you may be wondering. Well, in Part 2 you began thinking about prime numbers and modular arithmetic. You are probably aware that a major application of number theory is to the encryption of messages. I thought I’d share with you a fairly simple (and far from secure) method of encrypting a message.

To start with, we map the alphabet to $\{1, 2, 3, \ldots 26\}$ in a simple manner, setting $A = 1$, $B = 2$ down to $Z = 26$. We then do a lot of arithmetic modulo 26, which you’ll notice is not a prime number. This causes some complications because “zero” has factors: $2 \times 13 = 0$.

Our method involves encrypting more than one letter at a time. We write a string of letters as a column vector, and then multiply it by a suitable matrix, working modulo 26. This gives us an output vector of the same length, which is our encrypted string.

For simplicity, we choose a 2 × 2 matrix and in honour of one of your lecturers:

$$M = \begin{pmatrix} L & A \\ W & N \end{pmatrix} = \begin{pmatrix} 12 & 1 \\ 23 & 14 \end{pmatrix}.$$

The matrix M has determinant $12(14) - (1)(23) = 168 - 23 = 145 = 15 \bmod 26$. For the mapping to be a bijection, 15 has to be nonzero and coprime to 26, which it is. (I put a note on Piazza about this, which is what gave me the idea for this treasure hunt.) So for example, to encrypt the initials KB we form the vector $(11, 2)$ and then calculate

$$\begin{pmatrix} 12 & 1 \\ 23 & 14 \end{pmatrix} \begin{pmatrix} 11 \\ 2 \end{pmatrix} = \begin{pmatrix} 12(11) + 1(2) \\ 23(11) + 14(2) \end{pmatrix} = \begin{pmatrix} 134 \\ 281 \end{pmatrix} \bmod 26 = \begin{pmatrix} 4 \\ 21 \end{pmatrix} = \begin{pmatrix} D \\ U \end{pmatrix}.$$

So to encrypt a message we chop it up into two-letter strings and encrypt it two letters at a time. That’s all there is to it. To find the treasure you have to somehow **decrypt** the following message:

MSTMRAAAKPONNURAOMKPAZUYODBCONRTIYCVQGNMYXODBCUAZBKPQB

Good luck! Needless to say, this is entirely optional.