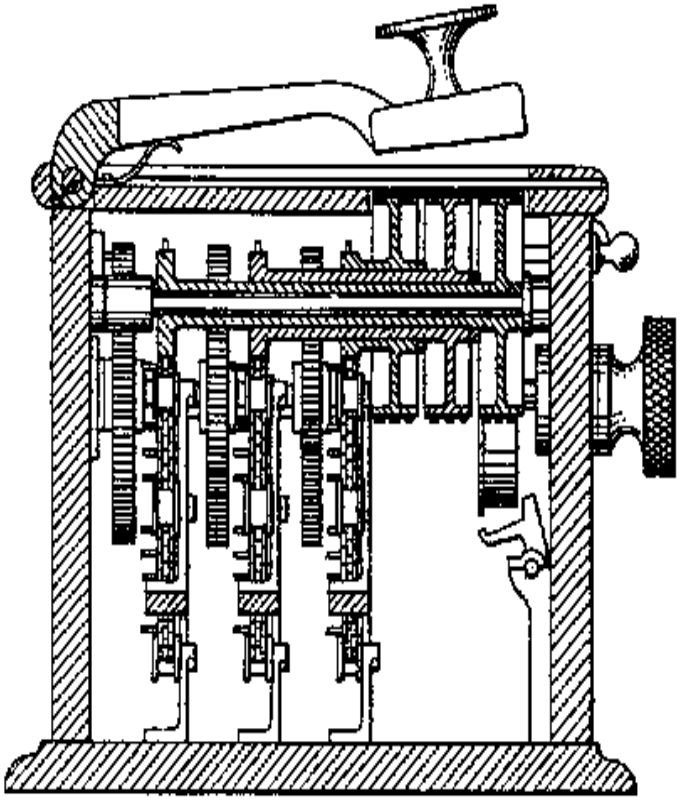
Text Reference: Section 4.1
Wikipedia Reference: Hill Cipher

Lab 5: Hill Substitution Ciphers

In this Lab, matrices are used to encode and decode messages. The ideas are due to Lester Hill in 1929. Hill's patented cipher machine appears in the image below.



Perhaps the simplest way to encode a message is to simply replace each letter of the alphabet with another letter. This is the method used in the **Cryptograms** often found in puzzle books or newspapers, and it is called a **substitution cipher**. Consider this cipher array for a substitution cipher:

```
AZ := "ABCDEFGHIJKLMNOPQRSTUVWXYZ": # The alphabet string
C1 := "YCWOGRDBPIZXKLNMTSEFHJAVUQ": # The cipher string
symbolVec:=s->Vector(26,i->convert(s[i],name)); # Convert a string
into a vector of symbols
mC1 := < 'source',symbolVec(AZ) | 'code',symbolVec(C1)>^+;
```

source	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
code	Y	C	W	O	G	R	D	B	P	I	Z	X	K	L	N	M	T	S	E	F	H	J	A	V	U	Q

To encode a message like VECTOR SPACE, encode each letter in turn and get JGWFNSEMYWG. Notice that the space between the words hasn't been encoded; this character could be added (as could any other punctuation or symbol) to the cipher array and encoded also. To decode a message, the decipher string is needed, which for our substitution cipher C1 := "YCWOGRDBPIZXKLNMTSEFHJAVUQ" is

```
D1 := cat(seq( AZ[SearchText( AZ[i], C1 )], i=1..26 ));
mD1 := < 'source',symbolVec(AZ) | 'message',symbolVec(D1)>^+;
```

$$D1 := \text{"WHBGSTEUJVMNPODIZFRQYXCLAK"}$$

$$\begin{bmatrix} code & A & B & C & D & E & F & G & H & I & J & K & L & M & N & O & P & Q & R & S & T & U & V & W & X & Y & Z \\ message & W & H & B & G & S & T & E & U & J & V & M & N & P & O & D & I & Z & F & R & Q & Y & X & C & L & A & K \end{bmatrix}$$

Thus the secret message FBGSNNEFGSWSNAEYFOYAL is easily revealed as THE ROOSTER CROWS AT DAWN using the decipher table above. The maple implementation is

```
AZ:="ABCDEFGHIJKLMNOPQRSTUVWXYZ": # The alphabet string
C1:="YCWOGRDBPIZXKLNMTSEFHJAVUQ": # The cipher string
D1:="WHBGSTEUJVMNPODIZFRQYXCLAK": # The decipher string
secret:="FBGSNNEFGSWSNAEYFOYAL";
cat(seq( D1[SearchText( secret[i], AZ )], i=1..length(secret) ));
```

A general decoding of a substituion cipher is made by the sample maple code

```
Decode:=msg->cat(seq( D1[SearchText( msg[i], AZ )], i=1..length(msg)
));
Decode("FBGSNNEFGSWSNAEYFOYAL");
```

"THEROOSTERCROWSATDAWN"

A major drawback of the substitution cipher is that it is very easy for a person to **crack** the code; that is, to determine the decipher array from an encoded message. If you have done a cryptogram before, you know how this is done: the relative frequency of letters in English is known, as are the frequencies of certain groups of letters like TH or ST. See Reference 3 (p.16 and p.19) for sample tables. Common short words like THE and OF also help the code cracker. To make the code harder to crack, groups of letters can be encoded at the same time. For example, consider splitting the above message into units of three letters each, although any length of block would be allowable. The message THE ROOSTER CROWS AT DAWN is thus converted to THE ROO STE RCR OWS ATD AWN. If the number of letters is not a multiple of three, the final set of letters can be padded with random letters; thus VECTOR SPACE could be split up as VEC TOR SPA CEX.

Since linear algebra will come in handy for the encoding process, replace each letter by its position in the alphabet. The positions are numbered from 0 to 25 for reasons which will shortly become apparent.

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

The batches of three letters can be converted into batches of three numbers, which can be thought of as **vectors**. For example, THE ROOSTER CROWS AT DAWN has become

$$\begin{bmatrix} T \\ H \\ E \end{bmatrix} \text{->} \begin{bmatrix} 19 \\ 7 \\ 4 \end{bmatrix}, \begin{bmatrix} R \\ O \\ O \end{bmatrix} \text{->} \begin{bmatrix} 17 \\ 14 \\ 14 \end{bmatrix}, \begin{bmatrix} S \\ T \\ E \end{bmatrix} \text{->} \begin{bmatrix} 18 \\ 19 \\ 4 \end{bmatrix}, \begin{bmatrix} R \\ C \\ R \end{bmatrix} \text{->} \begin{bmatrix} 17 \\ 2 \\ 17 \end{bmatrix}, \begin{bmatrix} O \\ W \\ S \end{bmatrix} \text{->} \begin{bmatrix} 14 \\ 22 \\ 18 \end{bmatrix},$$
$$\begin{bmatrix} A \\ T \\ D \end{bmatrix} \text{->} \begin{bmatrix} 0 \\ 19 \\ 3 \end{bmatrix}, \begin{bmatrix} A \\ W \\ N \end{bmatrix} \text{->} \begin{bmatrix} 0 \\ 22 \\ 13 \end{bmatrix}$$

In order to encode messages using the numbers instead of letters, an easy way of calculating using only the numbers 0 through 25 is needed. A way to do this is called **modular arithmetic**. Sums, differences, and products are calculated as normal, but if the result is larger than 25 or less than 0, it is replaced by the remainder left when the result is divided by 26. Thus only numbers from 0 to 25 result from this arithmetic. This form of arithmetic is denoted by stating that the calculations are to be done **modulo 26**, and by placing the symbol mod 26 after the calculation.

▼ Examples:

1 + 2 = 3 mod 26

13 x 2 = 0 mod 26, since 13*2 = 26 = 26(1) + 0

7 + 24 = 5 mod 26, since 7 + 24 = 31 = 26(1) + 5

10-15 = 21 mod 26, since 10 - 15 = -5 = 26(-1) + 21

Maple can do modular arithmetic as well: the Maple commands modp(m, n) and m mod n each return the value of m modulo n. Thus the command 7+24 mod 26 returns 5, just as in the third example above. Simple modular arithmetic can be done using this command. Just as numbers can be added and multiplied, vectors may be added and scalar multiplication may be performed:

▼ Examples:

$$\begin{bmatrix} 19 \\ 17 \\ 4 \end{bmatrix} + \begin{bmatrix} 14 \\ 22 \\ 18 \end{bmatrix} = \begin{bmatrix} 7 \\ 13 \\ 22 \end{bmatrix} (mod\ 26)$$

$$10 \begin{bmatrix} 14 \\ 22 \\ 18 \end{bmatrix} = \begin{bmatrix} 10 \\ 12 \\ 24 \end{bmatrix} (mod\ 26)$$

The mod and modp commands also can do modular arithmetic with vectors and matrices; for example, the command (Vector([19,17,4]) + Vector([14,22,18])) mod 26; returns the answer for the first example above. Parentheses are important to define the scope of modular arithmetic.

The set Z_{26}^3 is the set of all vectors with three elements drawn from the numbers 0 through 25. Addition of vectors and multiplication by scalars are defined as has been done above. The space Z_{26}^3 will be very useful for coding and decoding messages.

To encode a message a **key matrix** A is used, which in our case will be a 3x3 matrix. Multiplying a vector **v** in Z_{26}^3 by A will produce another vector A**v** which is also in Z_{26}^3, which may be interpreted as the encoded version of **v**.

▶ Example:

Now this method of encoding would be useless if there were not a way to decode the encoded messages.

Since the key matrix A does the encoding, it stands to reason that if $A = \begin{bmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{bmatrix}$ has an inverse A^{-1}, then A^{-1} should be the decoding matrix. While this is true, the inversion of A must be done in modular arithmetic. So for example the inverse of the key matrix $A = \begin{bmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{bmatrix}$ is $A^{-1} = \begin{bmatrix} 11 & 22 & 14 \\ 7 & 9 & 21 \\ 17 & 0 & 3 \end{bmatrix}$, because (see Question 1(d) below)

$$A^{-1}\ A = \begin{bmatrix} 11 & 22 & 14 \\ 7 & 9 & 21 \\ 17 & 0 & 3 \end{bmatrix} \begin{bmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \pmod{26}$$

Thus to decode the message ZRHMJHHKVVVWEXWHZIFRQ, the coded message is first split into three-letter units, which are then converted by alphabet translation into numbers and A-multiplication into vectors in Z_{26}^3, then these vectors are collected into a coded message matrix

$$C = \begin{bmatrix} 25 & 12 & 7 & 21 & 4 & 7 & 5 \\ 17 & 9 & 10 & 21 & 23 & 25 & 17 \\ 7 & 7 & 21 & 22 & 22 & 8 & 16 \end{bmatrix}$$

The computation

$$A^{-1}\ C = \begin{bmatrix} 11 & 22 & 14 \\ 7 & 9 & 21 \\ 17 & 0 & 3 \end{bmatrix} \begin{bmatrix} 25 & 12 & 7 & 21 & 4 & 7 & 5 \\ 17 & 9 & 10 & 21 & 23 & 25 & 17 \\ 7 & 7 & 21 & 22 & 22 & 8 & 16 \end{bmatrix} = \begin{bmatrix} 19 & 12 & 19 & 13 & 0 & 11 & 3 \\ 7 & 0 & 8 & 18 & 21 & 0 & 4 \\ 4 & 17 & 0 & 7 & 4 & 13 & 3 \end{bmatrix} \pmod{26}$$

gives the decoded message, which may be translated into letters as THEMARTIANSHAVELANDED, or THE MARTIANS HAVE LANDED. An automated way to do the translation is to concatenate the columns into a list, then apply the inverse translation from the alphabet.

```
    [19  12  19  13   0  11  3]
F:= [ 7   0   8  18  21   0  4]
    [ 4  17   0   7   4  13  3]
F_list:=[seq(seq(F[i,j],i=1..3),j=1..7)];
result:=map(n->AZ[n+1],F_list);
```

["T", "H", "E", "M", "A", "R", "T", "I", "A", "N", "S", "H", "A", "V", "E", "L", "A", "N", "D", "E", "D"]

The method which has been used to encode messages is called a **Hill substitution cipher**. The method was invented by mathematician Lester Hill and is reviewed in References 1 and 2. This form of cipher is harder to decode than the simple substitution cipher. Each letter is not encoded by the same letter over

and over; for example, in the above code for THE MARTIANS HAVE LANDED, the letter A appears four times and is encoded by J, V, E, and Z.

Since the inverse of the key matrix is needed to decode the encoded message, any prospective key matrix must be invertible. As has been seen in Chapter 3, the determinant of the matrix shows whether a given matrix is invertible. However, since calculations are being done in modular arithmetic, the following criterion must be used. It is given in Reference 3, page 114.

Fact:

An n x n matrix A is invertible modulo m if and only if det A $\neq$ 0 mod p for every prime divisor p of m.

Matrix A invertible modulo 26 means there is a matrix B such that AB = BA = I modulo 26. Thus a matrix A will be invertible modulo 26 (the inverse matrix B exists) if and only if det A $\neq 0$ mod 13 and also det A $\neq 0$ mod 2. **Warning**: If the determinant of matrix A is nonzero modulo 26, then the two relations det A $\neq 0$ mod 13 and det A $\neq 0$ mod 2 may or not hold, in short, A may not be invertible modulo 26!

Example:

Since det $\begin{bmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{bmatrix}$ = -1635, which is equal to 3 mod 13 and 1 mod 2, this matrix is invertible modulo 26 and may be used as a key matrix for a Hill substitution cipher.

Example:

Since det $\begin{bmatrix} 21 & 24 & 2 \\ 0 & 1 & 3 \\ 21 & 19 & 17 \end{bmatrix}$ = 630, which is equal to 6 mod 13 but 0 mod 2, this matrix is not invertible modulo 26 and may not be used as a key matrix.

Maple can calculate the determinant of a matrix using modular arithmetic. To compute the determinant of the first example matrix modulo 13, one can use the commands

```
A:=Matrix( [[3,10,20],[20,9,17],[9,4,17]] );
Determinant(A) mod 13.
```

Once it has been determined that a matrix A is invertible modulo 26, it may be used as a key matrix for a Hill substitution cipher. In order to decode this cipher the inverse of the key matrix A modulo 26 must be found. The same algorithm as presented in Section 2.2 of the text may be used: the matrix [A | I] is row reduced to [I | A^{-1}], but the reduction is done in modular arithmetic.

Row reduction can be a bit tricky here, because not all row operations are reversible when using mod 26

arithmetic. Scale operations are restricted to the reversible ones, namely scale row r by m provided 1/m exists modulo 26 (see the table at the end of the Lab for candidates m). Replacement operations are reversible modulo 26, because each element in Z_{26} has an additive inverse. Swap operations are reversible, modulo 26 having no impact.

The modulo 26 multiplication table at the end of this Lab will be helpful in checking modulo 26 arithmetic in the following example.

▼ Example:

It was shown above that the matrix A= $\begin{bmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{bmatrix}$ is invertible modulo 26 with inverse A^{-1} = $\begin{bmatrix} 11 & 22 & 14 \\ 7 & 9 & 21 \\ 17 & 0 & 3 \end{bmatrix}$.

To find the inverse, the algorithm begins with the matrix $G = \begin{bmatrix} 3 & 10 & 20 & 1 & 0 & 0 \\ 20 & 9 & 17 & 0 & 1 & 0 \\ 9 & 4 & 17 & 0 & 0 & 1 \end{bmatrix}$. Derivation details will be supplied for the modulo 26 reduced row echelon form of G, rref(G) = $\begin{bmatrix} 1 & 12 & 0 & 17 & 0 & 6 \\ 0 & 1 & 0 & 7 & 9 & 21 \\ 0 & 0 & 1 & 17 & 0 & 3 \end{bmatrix}$.

The first row of G must be scaled so that there is a 1 in the first pivot position. Solve 3x = 1 mod 26, answer x=9 (consult the multiplication table for modulo 26 at the end of this Lab), then scale row 1 by 9 to produce G1= $\begin{bmatrix} 1 & 12 & 24 & 9 & 0 & 0 \\ 20 & 9 & 17 & 0 & 1 & 0 \\ 9 & 4 & 17 & 0 & 0 & 1 \end{bmatrix}$. To eliminate the 20 below the first pivot, solve 20+3x=0 modulo 26, answer x=6. Then multiply row 1 by 6 and add it to row 2, giving the answer $G2 = \begin{bmatrix} 1 & 12 & 24 & 9 & 0 & 0 \\ 0 & 3 & 5 & 2 & 1 & 0 \\ 9 & 4 & 17 & 0 & 0 & 1 \end{bmatrix}$; likewise multiply row 1 by 17 and add it to row 3 to give $G3$

$= \begin{bmatrix} 1 & 12 & 24 & 9 & 0 & 0 \\ 0 & 3 & 5 & 2 & 1 & 0 \\ 0 & 0 & 9 & 23 & 0 & 1 \end{bmatrix}$. Multiply row 3 by 11 and add to row 2, then multiply row 3 by 6 and add to row 1, to produce G5=

$$\begin{bmatrix} 1 & 12 & 0 & 17 & 0 & 6 \\ 0 & 3 & 0 & 21 & 1 & 11 \\ 0 & 0 & 9 & 23 & 0 & 1 \end{bmatrix}$$. Replace row 1 by row 1 plus 22 times row 2 to get $G6 := \begin{bmatrix} 1 & 0 & 0 & 11 & 22 & 14 \\ 0 & 3 & 0 & 21 & 1 & 11 \\ 0 & 0 & 9 & 23 & 0 & 1 \end{bmatrix}$. Finally, scale rows 2 and 3 to give G8= $\begin{bmatrix} 1 & 12 & 0 & 17 & 0 & 6 \\ 0 & 1 & 0 & 7 & 9 & 21 \\ 0 & 0 & 1 & 17 & 0 & 3 \end{bmatrix}$.

Thus $A^{-1} = \begin{bmatrix} 11 & 22 & 14 \\ 7 & 9 & 21 \\ 17 & 0 & 3 \end{bmatrix}$ as noted above.

Maple can also do all of this. The package **laylinalg** can be used to do the row operations:

```
with(laylinalg):
G:=Matrix([[3,10,20,1,0,0],[20,9,17,0,1,0],[9,4,17,0,0,1]])
solve(3*x=1,x) mod 26; # Answer = 9, so scale by 9 modulo 26
G1:=scale(G,1,9) mod 26;
solve(20+3*x=0,x) mod 26; # solve 20+3x=0 modulo 26, answer x=6.
G2:=replace(G1,2,6,1) mod 26;
G3:=replace(G2,3,17,1) mod 26;
G4:=replace(G3,2,11,3) mod 26;
G5:=replace(G4,1,6,3) mod 26;
G6:=replace(G5,1,22,2) mod 26;
G7:=scale(G6,2,9) mod 26;
G8:=scale(G7,3,3) mod 26;
```

The command **`GaussianElimination( M ) mod n`** will reduce the matrix M to row echelon form while doing the arithmetic modulo n. In 2015 maple, commands **`ReducedRowEchelonForm( M ) mod n`** and **`MatrixInverse( M ) mod n are implemented.`** Also, **`1/M mod 26`** prints the inverse of M, modulo 26, without the LinearAlgebra package being loaded.

Problems to be Submitted:

Problem 1.

Compute the following modulo 26.

(a)

17+24

(b)

20 x 5

(c)

$$7\begin{bmatrix} 4 \\ 12 \\ 21 \end{bmatrix} - 3\begin{bmatrix} 14 \\ 5 \\ 16 \end{bmatrix}$$

(d)

$$\begin{bmatrix} 11 & 22 & 14 \\ 7 & 9 & 21 \\ 17 & 0 & 3 \end{bmatrix}\begin{bmatrix} 3 & 10 & 20 \\ 20 & 9 & 17 \\ 9 & 4 & 17 \end{bmatrix}$$

Problem 2.

Encode the phrase BUY TEN SHARES TOMORROW using the key matrix A from the Examples.

Problem 3.

Decode the phrase KSKBHXKDYRVTKRZTQE which was encoded using the key matrix A from the Examples.

Problem 4.

Determine whether the following matrices are invertible modulo 26.

(a)

$$\begin{bmatrix} 11 & 20 & 20 \\ 2 & 1 & 24 \\ 9 & 3 & 3 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 2 & 5 & 0 \\ 22 & 9 & 4 \\ 17 & 21 & 8 \end{bmatrix}$$

(c)

$$\begin{bmatrix} 3 & 1 & 24 \\ 20 & 11 & 25 \\ 12 & 4 & 19 \end{bmatrix}$$

Problem 5.

Consider the key matrix B = $\begin{bmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{bmatrix}$.

(a)

Encode the message MARY HAD A LITTLE LAMB using this key matrix.

(b)

Show that the matrix B is invertible.

(c)

Find B^{-1} modulo 26.

(d)

Decode the message FVRMTGJTJJRMULSIRGBEMRNVFRC which was encoded using the key matrix B.

Problem 6.

Consider the key matrix A = $\begin{bmatrix} 11 & 20 & 20 \\ 2 & 1 & 24 \\ 9 & 3 & 3 \end{bmatrix}$.

(a)

Encode the message RED SKY AT NIGHT using this key matrix.

(b)

Show that the matrix A is invertible.

(c)

Find A^{-1} modulo 26.

(d)

Decode the message IWGEJLFWRBUEUOWBHPZMLMXNXUBOEUAHG which was encoded using the key matrix A.

References

1.

Hill, Lester. "Cryptography in an Algebraic Alphabet." *American Mathematical Monthly*, June-July 1929, pp. 306-312.

2.

Hill, Lester. "Concerning Certain Linear Transformation Apparatus of Cryptography." *American Mathematical Monthly*, March 1931, pp. 135-154.

3.

Konheim, Alan G. *Cryptography: A Primer*. New York: John Wiley and Sons, 1981.

4.

Bretscher, Otto. *Linear Algebra with Applications, 4/E.* New York: Pearson, 2008. ISBN 0136009263, 9780136009269.

Multiplication Table (mod 26)

[[, 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],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[1, 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, 0],
[2, 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 0],
[3, 0, 3, 6, 9, 12, 15, 18, 21, 24, 1, 4, 7, 10, 13, 16, 19, 22, 25, 2, 5, 8, 11, 14, 17, 20, 23, 0],
[4, 0, 4, 8, 12, 16, 20, 24, 2, 6, 10, 14, 18, 22, 0, 4, 8, 12, 16, 20, 24, 2, 6, 10, 14, 18, 22, 0],
[5, 0, 5, 10, 15, 20, 25, 4, 9, 14, 19, 24, 3, 8, 13, 18, 23, 2, 7, 12, 17, 22, 1, 6, 11, 16, 21, 0],
[6, 0, 6, 12, 18, 24, 4, 10, 16, 22, 2, 8, 14, 20, 0, 6, 12, 18, 24, 4, 10, 16, 22, 2, 8, 14, 20, 0],
[7, 0, 7, 14, 21, 2, 9, 16, 23, 4, 11, 18, 25, 6, 13, 20, 1, 8, 15, 22, 3, 10, 17, 24, 5, 12, 19, 0],
[8, 0, 8, 16, 24, 6, 14, 22, 4, 12, 20, 2, 10, 18, 0, 8, 16, 24, 6, 14, 22, 4, 12, 20, 2, 10, 18, 0],
[9, 0, 9, 18, 1, 10, 19, 2, 11, 20, 3, 12, 21, 4, 13, 22, 5, 14, 23, 6, 15, 24, 7, 16, 25, 8, 17, 0],
[10, 0, 10, 20, 4, 14, 24, 8, 18, 2, 12, 22, 6, 16, 0, 10, 20, 4, 14, 24, 8, 18, 2, 12, 22, 6, 16, 0],
[11, 0, 11, 22, 7, 18, 3, 14, 25, 10, 21, 6, 17, 2, 13, 24, 9, 20, 5, 16, 1, 12, 23, 8, 19, 4, 15, 0],
[12, 0, 12, 24, 10, 22, 8, 20, 6, 18, 4, 16, 2, 14, 0, 12, 24, 10, 22, 8, 20, 6, 18, 4, 16, 2, 14, 0],
[13, 0, 13, 0, 13, 0, 13, 0, 13, 0, 13, 0, 13, 0, 13, 0, 13, 0, 13, 0, 13, 0, 13, 0, 13, 0],
[14, 0, 14, 2, 16, 4, 18, 6, 20, 8, 22, 10, 24, 12, 0, 14, 2, 16, 4, 18, 6, 20, 8, 22, 10, 24, 12, 0],
[15, 0, 15, 4, 19, 8, 23, 12, 1, 16, 5, 20, 9, 24, 13, 2, 17, 6, 21, 10, 25, 14, 3, 18, 7, 22, 11, 0],
[16, 0, 16, 6, 22, 12, 2, 18, 8, 24, 14, 4, 20, 10, 0, 16, 6, 22, 12, 2, 18, 8, 24, 14, 4, 20, 10, 0],
[17, 0, 17, 8, 25, 16, 7, 24, 15, 6, 23, 14, 5, 22, 13, 4, 21, 12, 3, 20, 11, 2, 19, 10, 1, 18, 9, 0],
[18, 0, 18, 10, 2, 20, 12, 4, 22, 14, 6, 24, 16, 8, 0, 18, 10, 2, 20, 12, 4, 22, 14, 6, 24, 16, 8, 0],
[19, 0, 19, 12, 5, 24, 17, 10, 3, 22, 15, 8, 1, 20, 13, 6, 25, 18, 11, 4, 23, 16, 9, 2, 21, 14, 7, 0],
[20, 0, 20, 14, 8, 2, 22, 16, 10, 4, 24, 18, 12, 6, 0, 20, 14, 8, 2, 22, 16, 10, 4, 24, 18, 12, 6, 0],
[21, 0, 21, 16, 11, 6, 1, 22, 17, 12, 7, 2, 23, 18, 13, 8, 3, 24, 19, 14, 9, 4, 25, 20, 15, 10, 5, 0],
[22, 0, 22, 18, 14, 10, 6, 2, 24, 20, 16, 12, 8, 4, 0, 22, 18, 14, 10, 6, 2, 24, 20, 16, 12, 8, 4, 0],
[23, 0, 23, 20, 17, 14, 11, 8, 5, 2, 25, 22, 19, 16, 13, 10, 7, 4, 1, 24, 21, 18, 15, 12, 9, 6, 3, 0],
[24, 0, 24, 22, 20, 18, 16, 14, 12, 10, 8, 6, 4, 2, 0, 24, 22, 20, 18, 16, 14, 12, 10, 8, 6, 4, 2, 0],
[25, 0, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0],
[26, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]

The maple commands used to create this table are:

```
interface(rtablesize=28):
M2:=Matrix(27,27,(i,j)->(i-1)*(j-1) mod 26);
M3:=Vector(27,i->i-1);
M4:=<M3|M2>;
row1:=<'x',M3>^+;
tableMultiplyMod26:=<row1,M4>;
```