I'll use the built-in package "linalg".

There is a newer package "LinearAlgebra". You can find out about that by entering the command for help, "? LinearAlgebra".

When you enter the command "with [some package]", you get a list of the built-in operations that come in that package.

```maple
> with(linalg);
Warning, the protected names norm and trace have been redefined and unprotected

[BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint,
angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim,
colspace, colspan, companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det,
diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential,
extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad,
hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith,
issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly,
mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix,
randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith,
stackmatrix, submatrix, subvector, sumbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose,
vandermonde, vecpotent, vectdim, vector, wronskian]
```

```maple
> M:=matrix([[1,2,3], [4,5,6], [9,8,7]]);
M :=

[ 1 2 3 ]
[ 4 5 6 ]
[ 9 8 7 ]

The determinant and (if M is invertible) inverse of M are built-in operations.

```maple
> det(M);
0
```

Oops. Since det(M) = 0, M is not invertible - there is no inverse matrix for M. If we put M in row-reduced echelon form, we can see that.

```maple
> rref(M);
```

Let's change M to get an invertible matrix. We will change the entry in row 3, column 2.

\[
M[3,2] := 5;
\]

\[
M
\]

\[
\begin{bmatrix}
1 & 0 & -1 \\
0 & 1 & 2 \\
0 & 0 & 0
\end{bmatrix}
\]

\[
\text{det}(M);
\]

\[-18\]

That's better. Now M is invertible.

\[
\text{inverse}(M);
\]

\[
\begin{bmatrix}
-5 & -1 & 1 \\
18 & 18 & 6 \\
-13 & 10 & -1 \\
9 & 9 & 3 \\
25 & -13 & 1 \\
18 & 18 & 6
\end{bmatrix}
\]

Note that Maple calculated the inverse in as much time as it took me to hit the "enter" key.

If we want decimals, use the extra command "evalf", which means "evaluate as floating-point numbers".

\[
evalf(M);
\]

Sometimes Maple give back cryptic, annoying replies like this. I wanted the matrix of numbers, and Maple didn't feel like writing all that. So let's force it by the command "evalm", which means "evaluate as a matrix", and should result in a printout of the matrix.

\[
evalm(evalf(inverse(M)))
\]

Here is how to multiply matrices.
Since $M$ has 3 columns, we can multiply $M$ by any matrix with 3 rows.

```maple
> N:=matrix([[2,0], [-1,3], [1,4]]);
N :=
[  2   0  ]
[     -1  3 ]
[     1   4 ]
```

To indicate matrix multiplication, you don't just write $M*N$. If I try that, here is the reprimand.

```maple
> evalm(M*N);
Error, (in evalm/evaluate) use the &* operator for matrix/vector multiplication
```

```maple
> evalm(M &* N);
            [  3   18 ]
            [     9  39 ]
            [ 20  43 ]
```

Finally, here is how you can use Maple to solve a system of linear equations.

We use the command "solve". We use {...} notation to denote a set of equations. We need to tell the computer what are the variables. This is because the computer could also solve the system if we had some parameters instead of the numerical coefficients.

```maple
> solve({2*x+3*y=4, 5*x-7*y=11}, {x,y});
{ y = -29, x = 61 }  
```

If the system is inconsistent, we get no reply; Maple just won't do it.

```maple
> solve({2*x+3*y=4, 2*x+3*y=5}, {x,y});
```

If the system has infinitely many solutions, we would get something like

```maple
> solve({2*x+3*y=4, 4*x+6*y=8}, {x,y});
{ x = 2 - 3/2*y, y = y }
```

That is Maple's way of telling us that there are infinitely many solutions: let $y$ be anything, and then take $x=2 - (3/2)y$. 

I hope you find this useful as a way to check your various calculations in linear algebra.