```
> restart; with(linalg); #We consider Problem 5.5.2 from [JH]
```

[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]

```
> for i from 1 to 6 do 3^i mod 7; od; #we check that 3 is primitive
  in F_7
```

$$3$$
$$2$$
$$6$$
$$4$$
$$5$$
$$1$$

```
> G:=matrix(4,6,[]):
> for i from 1 to 4 do
>     for j from 1 to 6 do
>         G[i,j]:=3^((i-1)*(j-1)) mod 7:
>     od:
> od: #A generator matrix, using the polynomials 1,x,x^2,x^3
> evalm(G);
```

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 2 & 6 & 4 & 5 \\ 1 & 2 & 4 & 1 & 2 & 4 \\ 1 & 6 & 1 & 6 & 1 & 6 \end{bmatrix}$$

```
> #We want to have a generator matrix in systematic form. We should
  consider polynomials that interpolate our points
> GG:=matrix(4,6,[]):
> p1:=x->(x-3)*(x-2)*(x-6) mod 7; p1(1);
```

$$p1 := x \rightarrow (x-3)\ (x-2)\ (x-6)\ \textbf{mod}\ 7$$

$$4$$

```
> p3:=x->(x-1)*(x-2)*(x-6) mod 7; p3(3);
```

$$p3 := x \to (x-1)\,(x-2)\,(x-6) \ \mathbf{mod}\ 7$$

$$1$$

```
> p2:=x->(x-1)*(x-3)*(x-6) mod 7; p2(2);
```

$$p2 := x \to (x-1)\,(x-3)\,(x-6) \ \mathbf{mod}\ 7$$

$$4$$

```
> p6:=x->(x-1)*(x-3)*(x-2) mod 7; p6(6);
```

$$p6 := x \to (x-1)\,(x-3)\,(x-2) \ \mathbf{mod}\ 7$$

$$4$$

```
> #We should divide p1, p2 and p6 by 4
> 4^(-1) mod 7;
```

$$2$$

```
> #or multiply times 2
> p1:=x->2*(x-3)*(x-2)*(x-6) mod 7; p1(1);
  p2:=x->2*(x-1)*(x-3)*(x-6) mod 7; p2(2);
  p6:=x->2*(x-1)*(x-3)*(x-2) mod 7; p6(6);
```

$$p1 := x \to 2\,(x-3)\,(x-2)\,(x-6) \ \mathbf{mod}\ 7$$

$$1$$

$$p2 := x \to 2\,(x-1)\,(x-3)\,(x-6) \ \mathbf{mod}\ 7$$

$$1$$

$$p6 := x \to 2\,(x-1)\,(x-3)\,(x-2) \ \mathbf{mod}\ 7$$

$$1$$

```
> for j from 1 to 6 do GG[1,j]:=p1(3^(j-1)): od:
> for j from 1 to 6 do GG[2,j]:=p3(3^(j-1)): od:
> for j from 1 to 6 do GG[3,j]:=p2(3^(j-1)): od:
> for j from 1 to 6 do GG[4,j]:=p6(3^(j-1)): od:
> evalm(GG);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 6 & 2 \\ 0 & 1 & 0 & 0 & 2 & 2 \\ 0 & 0 & 1 & 0 & 2 & 5 \\ 0 & 0 & 0 & 1 & 5 & 6 \end{bmatrix}$$

```
> #Minimum distance.
> n:=6; k:=4;
```

$$n := 6$$

$$k := 4$$

```
> d:=n + 1 - k;
```

$$d := 3$$

```
> #Codeword from f(x)=x^3 + x?

> #we can use the generator matrix
> c:=evalm([0,1,0,1]&*G mod 7);
```

$$c := \begin{bmatrix} 2 & 9 & 3 & 12 & 5 & 11 \end{bmatrix}$$

```
> for i to 6 do c[i]:=c[i] mod 7 od: #mod 7 does not work very well
  ...
> evalm(c);
```

$$\begin{bmatrix} 2 & 2 & 3 & 5 & 5 & 4 \end{bmatrix}$$

```
> #or we can evaluate the polynomial x+ x^3
> f:=x->x + x^3;
```

$$f := x \to x + x^3$$

```
> c:=matrix(1,6,[]):
> for j from 1 to 6 do c[1,j]:=f(3^(j-1)) mod 7: od:evalm(c);
```

$$\begin{bmatrix} 2 & 2 & 3 & 5 & 5 & 4 \end{bmatrix}$$

```
> #or we can add row 2 and 4 of the generator matrix
> c:=evalm(row(G,2)+row(G,4)): for i from 1 to 6 do c[i]:=c[i] mod
  7 od: evalm(c);
```

$$\begin{bmatrix} 2 & 2 & 3 & 5 & 5 & 4 \end{bmatrix}$$

```
> #We should add 2 to position 3
> r:=c; r[3]:=r[3]+2: evalm(r);
```

$$r := c$$

$$\begin{bmatrix} 2 & 2 & 5 & 5 & 5 & 4 \end{bmatrix}$$

```
> #and try to correct this error
> #l_0, l_1
> t:=floor((d-1)/2);
```

$$t := 1$$

```
> l0:=n-1-t;
```

$$l0 := 4$$

```
> l1:=n-1-t-(k-1);
```

$$l1 := 1$$

```
> #So we have 7 coefficients and 6 equations
> A:=matrix(6,7,[]):
> for i from 1 to 6 do
>     for j from 1 to l0+1 do
>         A[i,j]:=3^((i-1)*(j-1)) mod 7:
>     od:
> od:
```

```
> for i from 1 to 6 do
>     for j from l0+2 to l0+l1+2 do
>         A[i,j]:=r[i]*3^((i-1)*(j-(l0+2))) mod 7:
>     od:
> od:
> eval(A);
```

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 \\ 1 & 3 & 2 & 6 & 4 & 2 & 6 \\ 1 & 2 & 4 & 1 & 2 & 5 & 3 \\ 1 & 6 & 1 & 6 & 1 & 5 & 2 \\ 1 & 4 & 2 & 1 & 4 & 5 & 6 \\ 1 & 5 & 4 & 6 & 2 & 4 & 6 \end{bmatrix}$$

```
> b:=vector([0,0,0,0,0,0]);
```

$$b := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> Q:=Linsolve(A, b) mod 7;
```

$$Q := \begin{bmatrix} 0 & 2\_t_7 & 6\_t_7 & 2\_t_7 & 6\_t_7 & 5\_t_7 & \_t_7 \end{bmatrix}$$

```
> Q0:=2*x+6*x^2+2*x^3+6*x^4;
```

$$Q0 := 2\,x + 6\,x^2 + 2\,x^3 + 6\,x^4$$

```
> Q1:=5+x;
```

$$Q1 := 5 + x$$

```
> Divide(-Q0,Q1,'g') mod 7;
```

true

```
> g; #so we get the codeword
```

$$x + x^3$$

```
> H:=matrix(n-k,n,[]):
> for i from 1 to n-k do
>     for j from 1 to n do
>         H[i,j]:=3^((i)*(j-1)) mod 7:
>     od:
> od: evalm(H); #Parity check matrix
```

$$\begin{bmatrix} 1 & 3 & 2 & 6 & 4 & 5 \\ 1 & 2 & 4 & 1 & 2 & 4 \end{bmatrix}$$