1 art;maple_mode(0);cas_setup(0,0,0,1,0,1e-10,25,[1,50,0,25],0,0,0);//radians,pas de cmplx, pas de Sqrt

[[], Warning: some commands like subs might change arguments order , 0, 0, 0, 1, 0, 0.9999999999]

2 diag(seq(1,4)); diag(1$4);

$$\left(\begin{bmatrix} 1, & 0, & 0, & 0 \\ 0, & 1, & 0, & 0 \\ 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix}, \begin{bmatrix} 1, & 0, & 0, & 0 \\ 0, & 1, & 0, & 0 \\ 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix} \right)$$

3 diag(seq(j,j=1..4));

$$\begin{bmatrix} 1, & 0, & 0, & 0 \\ 0, & 2, & 0, & 0 \\ 0, & 0, & 3, & 0 \\ 0, & 0, & 0, & 4 \end{bmatrix}$$

4 A:=matrix(4,4)+1;v:=[seq(1,j=1..4)];

$$\left(\begin{bmatrix} 1, & 0, & 0, & 0 \\ 0, & 1, & 0, & 0 \\ 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix}, [1, 1, 1, 1] \right)$$

5 A*v;// Attention il retourne une ligne,pour xcas les vecteurs sont en ligne

$$[1, 1, 1, 1]$$

6 purge(a);

No such variable a

7 f:=(ii,j)->if (ii=j) then 0 else if (ii<j) then a[ii,j] else -a[j,ii] fi;fi;

// Warning: a, declared as global variable(s)
// End defining f

```
                    if ((ii==j)) {
                       0;
                     }
                     else {
                       if (ii<j) {
                          a[ii,j];
                       }
                       else {
                         -(a[j,ii]);
                       }; ;
( ii, j )-> }; 
```

8 matrix(4,4,f); d:=det(matrix(4,4,f)):;

$$\left(\begin{bmatrix} 0, & a[(0, 1)], & a[(0, 2)], & a[(0, 3)] \\ -(a[(0, 1)]), & 0, & a[(1, 2)], & a[(1, 3)] \\ -(a[(0, 2)]), & -(a[(1, 2)]), & 0, & a[(2, 3)] \\ -(a[(0, 3)]), & -(a[(1, 3)]), & -(a[(2, 3)]), & 0 \end{bmatrix}, \text{Done} \right)$$

9 factor(d);//c'est toujours un carre

$$((a[(0, 1)]) \cdot (a[(2, 3)]) - (a[(0, 2)]) \cdot (a[(1, 3)]) + (a[(0, 3)]) \cdot (a[(1, 2)]))^2$$

10 M:=matrix(8,8,f);

0,	a[(0, 1)],	a[(0, 2)],	a[(0, 3)],	a[(0, 4)],	a[(0, 5)],	a[(0, 6)],	a
-(a[(0, 1)]),	0,	a[(1, 2)],	a[(1, 3)],	a[(1, 4)],	a[(1, 5)],	a[(1, 6)],	a
-(a[(0, 2)]),	-(a[(1, 2)]),	0,	a[(2, 3)],	a[(2, 4)],	a[(2, 5)],	a[(2, 6)],	a
-(a[(0, 3)]),	-(a[(1, 3)]),	-(a[(2, 3)]),	0,	a[(3, 4)],	a[(3, 5)],	a[(3, 6)],	a
-(a[(0, 4)]),	-(a[(1, 4)]),	-(a[(2, 4)]),	-(a[(3, 4)]),	0,	a[(4, 5)],	a[(4, 6)],	a
-(a[(0, 5)]),	-(a[(1, 5)]),	-(a[(2, 5)]),	-(a[(3, 5)]),	-(a[(4, 5)]),	0,	a[(5, 6)],	a
-(a[(0, 6)]),	-(a[(1, 6)]),	-(a[(2, 6)]),	-(a[(3, 6)]),	-(a[(4, 6)]),	-(a[(5, 6)]),	0,	a
-(a[(0, 7)]),	-(a[(1, 7)]),	-(a[(2, 7)]),	-(a[(3, 7)]),	-(a[(4, 7)]),	-(a[(5, 7)]),	-(a[(6, 7)]),	0

M

11 Methode type pivot, contre m\'ethode de Laplace,... mais ici la matrice est antisymetrique.

12 d1:=det(M):;

Evaluation time: 5.88

Done

M

13 d2:=det_minor(M):;

Done

M

14 normal(d1-d2);

Evaluation time: 2.8

0

M

15 ----------Illustration de la reduction de Jordan----------
Construction de l'exemple: on veut une reponse de ce type:

16 f:=(ii,j)->if(ii=j-1) then 1 else 0 fi;

// Success
// End defining f

```
if ((ii==(j-1))) {
  1;
}
else {
  0;
}
( ii, j )-> };
```

M

17 J:=matrix(8,8,f);//forme classique d'ordre 8.

0,	1,	0,	0,	0,	0,	0,	0
0,	0,	1,	0,	0,	0,	0,	0
0,	0,	0,	1,	0,	0,	0,	0
0,	0,	0,	0,	1,	0,	0,	0
0,	0,	0,	0,	0,	1,	0,	0
0,	0,	0,	0,	0,	0,	1,	0
0,	0,	0,	0,	0,	0,	0,	1
0,	0,	0,	0,	0,	0,	0,	0

18 J[2,3]:=0:;J[5,6]:=0:;J;//2 blocs d'ordre 3 et un d'ordre 2.

$$\left(\text{Done}, \quad \text{Done}, \quad \begin{bmatrix} 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \end{bmatrix} \right)$$

M

19 On veut faire un changement de base simple. Ex det=1 pour garder des coeffs entiers.
on cree une transvection:

20 f:=(ii,j)->if(ii=j) then 1 else 0 fi;

// Success
// End defining f

```
if ((ii==j)) {
  1;
}
else {
  0;
};
```

(ii, j)> M

21 T:=proc(ii,j,a)
local A;
A:=matrix(8,8,f):;A[ii-1,j-1]:=a;A;
end proc;

// Warning: f, declared as global variable(s)
// End defining T

```
(ii,j,a)->
{ local A;
  A:=matrix(8,8,f);
  A[ii-1,j-1]:=a;
  A;
}
```

M

22 B:=matrix(8,8,b);

$$\begin{bmatrix} b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \end{bmatrix}$$

M

23 faire Li<-Li+aLj c'est multiplier a gauche par T(i,j,a).
Par exemple L3<- L3+aL2 c'est multiplier a GAUCHE par: T(3,2,a);

24 T(3,2,a)*B;

$$\begin{bmatrix} b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \\ a*b+b, & a*b+b, & a*b+b, & a*b+b, & a*b+b, & a*b+b, & a*b+b, & a*b+b \\ b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \\ b, & b, & b, & b, & b, & b, & b, & b \end{bmatrix}$$

M

25 En revanche: C2<-C2+aC3 c'est multplier a DROITE par T(3,2,a)

26 B*T(3,2,a);

$$\begin{bmatrix} b, & a*b+b, & b, & b, & b, & b, & b, & b \\ b, & a*b+b, & b, & b, & b, & b, & b, & b \\ b, & a*b+b, & b, & b, & b, & b, & b, & b \\ b, & a*b+b, & b, & b, & b, & b, & b, & b \\ b, & a*b+b, & b, & b, & b, & b, & b, & b \\ b, & a*b+b, & b, & b, & b, & b, & b, & b \\ b, & a*b+b, & b, & b, & b, & b, & b, & b \\ b, & a*b+b, & b, & b, & b, & b, & b, & b \end{bmatrix}$$

M

27 Remarquer que l'inverse de T(i,j,a) est T(i,j,-a)

28 T(3,2,a)^(-1);

$$\begin{bmatrix} 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & -a, & 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1 \end{bmatrix}$$

M

29 Donc conjuguer par T(i,j,a) c'est faire:
Li <- Li+aLj et Cj <- Cj-aCi

30 P:=T(6,7,2)*T(4,5,1)*T(3,2,2)*T(1,2,1):;

Done

M

31 P,P^(-1);

$$\left(\begin{bmatrix} 1, & 1, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 2, & 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 1, & 1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 1, & 2, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1 \end{bmatrix}, \begin{bmatrix} 1, & -1, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & -2, & 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 1, & -1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 1, & -2, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1 \end{bmatrix} \right)$$

M

32 Donc faire a l'ordinateur:

33 N:=P*J*P^(-1):;

Done

34 est identique a faire a la main a partir de J:
L1 <- L1+L2 ; C2<-C2 - C1 puis
L3 <- L3+2L2 ; C2 <- C2 -2C3
L4 <- L4+L5 ; C5 <- C5 -C4
L6 <- L6 +2 L7; C7 <- C7 -2 C6
On a maintenant trouve un bel exercice: Trouver la forme de
jordan de N et une matrice de passage pour l'obtenir.
On calcule N^2 et son noyau.

35 N,N^2;

$$\left(\begin{bmatrix} 0, & -1, & 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & -2, & 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & -4, & 2, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 1, & 1, & -2, & 0 \\ 0, & 0, & 0, & 0, & 0, & 1, & -2, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 2 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \end{bmatrix}, \begin{bmatrix} 0, & -2, & 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 1, & -2, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \end{bmatrix}\right)$$

36 N2:=nullspace(N^2);

$$\begin{bmatrix} -1, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & \frac{-1}{2}, & -1, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & -1, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & -1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & -2, & -1, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & -1 \end{bmatrix}$$

37 on choisit a et b independants modulo ker N^2 (qui est aussi im
N). (attention a et b hors de ker N^2 est insuffisant).

38 a:=[0,0,1,0,0,0,0,0]:;b:=[0,0,0,0,0,1,0,0]:;

(Done , Done)

39 rank(matrix([op(N2),a,b]));// doit etre dim ker N^2 +2.

8

40 N1:=nullspace(N);

$$\begin{bmatrix} -1, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & -1, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & -2, & -1, & 0 \end{bmatrix}$$

41 dim ker N^2 -dim ker N= 6-3=2+1 donc N.a,N.b doit etre complete par c
tq N.a,N.b,c indep modulo ker N. Par exemple on prend celui la:

42 c:=[0,0,0,0,0,0,0,1]:;

Done

43 On verifie qu'il convient:

44 rank(matrix([op(N1),N*a,N*b,c]));

6

45 dim ker N - dim ker N^0=3 c'est donc engendr\'e par
N^2.a,N^2.b,N.c. Il n'y a plus rien a faire, et l'on prend
la base suivante: (Attention pour xcas N*a... sont des lignes, on
prend donc la transposee)

46 Q:=transpose(matrix([(N^2)*a,N*a,a,(N^2)*b,N*b,b,N*c,c]));

$$\begin{bmatrix} 1, & 1, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 2, & 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 1, & 1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 1, & 2, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1 \end{bmatrix}$$

47 On sait maintenant que Q^(-1)*N*Q doit donner J. verifification:

48 Q^(-1)*N*Q;

$$\begin{bmatrix} 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \end{bmatrix}$$

49 ----------

50 Il s'agit donc de trouver les bases qui jordanisent l'endomorphisme.
On a N^3=0.
Il nous faut d'abord pour les 2 blocs de taille 3, une base de kerN^3/ker N^2 que l'on remonte ensuite dans ker N^3 (Choix de a et b dans la question precedente). Comme ker N^3/ker N^2 est de dimension 2, on a deja: card(GL2).(card ker N^2)^2. qui vaut: (p^2-1)*(p^2-p)*(p^6)^2. reste le choix d'un vecteur non nul de ker N^2/(ker N +Im N) que l'on remonte. ker N+im N est de dim 3+2. On a donc p-1 choix dans le quotient, soit (p-1)*p^5 choix pour le dernier vecteur a choisir (c dans la question precedente).

51 cardstab:=(p^2-1)*(p^2-p)*p^12*(p-1)*p^5

$$(p^2-1)\cdot(p^2-p)\cdot p^{12}\cdot(p-1)\cdot p^5$$

52 Autre methode: On commence par choisir les vecteurs de base qui sont dans le noyau. On prend une base de Im N^2 qui est de dim 2, soit (p^2-1)*(p^2-p) puis un vecteur du noyau hors de Im N^2. soit (p^3-p^2) choix. On cherche alors des antecedants a ces vecteurs. (p^3 choix pour l'antecedant d'un vecteur)

53 cardstab2:=(p^2-1)*(p^2-p)*(p^3-p^2)*(p^3)^3*(p^3)^2;

$$(p^2-1)\cdot(p^2-p)\cdot(p^3-p^2)\cdot p^9\cdot p^6$$

54 normal(cardstab-cardstab2);

$$0$$

55 cardGL8:=product((p^8-p^ii),ii=0..7);

$$(p^8-1)\cdot(p^8-p)\cdot(p^8-p^2)\cdot(p^8-p^3)\cdot(p^8-p^4)\cdot(p^8-p^5)\cdot(p^8-p^6)\cdot(p^8-p^7)$$

56 cardorbite:=normal(cardGL8/cardstab);

$$p^{42}+p^{41}+p^{40}-p^{38}-2\cdot p^{37}-3\cdot p^{36}-3\cdot p^{35}-3\cdot p^{34}-p^{33}+p^{32}+4\cdot p^{31}+5\cdot p^{30}+6\cdot p^{29}+5\cdot p^{28}+3\cdot p$$

57

58 ----------Exercice---syntaxe et combinatoire des matrice nilpotentes----------

```
59 n:=5;purge(a);
```

$$(5, [0, 0, 1, 0, 0, 0, 0, 0])$$

```
60 A:=matrix(n,n,(ii,j)->a[n*(ii)+j]);
// Warning: a,n, declared as global variable(s)
```

$$\begin{bmatrix} a[0], & a[1], & a[2], & a[3], & a[4] \\ a[5], & a[6], & a[7], & a[8], & a[9] \\ a[10], & a[11], & a[12], & a[13], & a[14] \\ a[15], & a[16], & a[17], & a[18], & a[19] \\ a[20], & a[21], & a[22], & a[23], & a[24] \end{bmatrix}$$

```
61 l:=[seq(a[ii],ii=0..n^2-1)];
```

[a[0], a[1], a[2], a[3], a[4], a[5], a[6], a[7], a[8], a[9], a[10], a[11], a[12], a[13], a[14], a[15],

```
62 Pour convertir la liste l en une matrice a n colonne:
63 A:=list2mat(l,n);
```

$$\begin{bmatrix} a[0], & a[1], & a[2], & a[3], & a[4] \\ a[5], & a[6], & a[7], & a[8], & a[9] \\ a[10], & a[11], & a[12], & a[13], & a[14] \\ a[15], & a[16], & a[17], & a[18], & a[19] \\ a[20], & a[21], & a[22], & a[23], & a[24] \end{bmatrix}$$

```
64 J:=matrix(n,n)::J[0,1]:=1::J[2,3]:=1::J;
```

$$\left(\text{Done}, \ \text{Done}, \ \text{Done}, \ \begin{bmatrix} 0, & 1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \end{bmatrix} \right)$$

```
65 L:=mat2list(A*J-J*A);
```

[-(a[5]), a[0]-(a[6]), -(a[7]), a[2]-(a[8]), -(a[9]), 0, a[5], 0, a[7], 0, -(a[15]), a[10]-(a[16]), -(a[17]

```
66 C:=matrix(n^2,n^2,(ii,j)->diff(L[ii],a[j]));
// Warning: L,a, declared as global variable(s)
```

$$\begin{bmatrix}
0, & 0, & 0, & 0, & 0, & -1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
1, & 0, & 0, & 0, & 0, & 0, & -1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 0, & 0, & -1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & -1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & -1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & -1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & -1, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & -1, & 0, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0, & -1, & 0, & 0, & 0, & 0, & 0 \\
0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0
\end{bmatrix}$$

```
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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

67 n^2-rank(C); //est la dim du commutateur de J.

13

68 En mode xcas c'est transparent, on travaille avec une matrice a coefficients modulaire avec l'instruction usuelle. C'est ici que le mode maple est tres desavantageux par rapport au mode xcas.

69 ker([[3%2,5%2],[1%2,1%2]]);

[1 % 2 , 1 % 2]

70 com:=ker(C%2); // C%2 donne la classe de C dans Mn(Z/2Z)

```
0 % 2 , 1 % 2 , 0 % 2 , 0,      0,      0 % 2 , 0,      0 % 2 , 0,      0 % 2 , 0 % 2 , 0,      0 % 2 , 0,
0 % 2 , 0,      0 % 2 , 1 % 2 , 0,      0 % 2 , 0,      0 % 2 , 0,      0 % 2 , 0 % 2 , 0,      0 % 2 , 0,
0 % 2 , 0,      0 % 2 , 0,      1 % 2 , 0 % 2 , 0,      0 % 2 , 0,      0 % 2 , 0 % 2 , 0,      0 % 2 , 0,
1 % 2 , 0,      0 % 2 , 0,      0,      0 % 2 , 1 % 2 , 0 % 2 , 0,      0 % 2 , 0 % 2 , 0,      0 % 2 , 0,
0 % 2 , 0,      1 % 2 , 0,      0,      0 % 2 , 0,      0 % 2 , 1 % 2 , 0 % 2 , 0 % 2 , 0,      0 % 2 , 0,
0 % 2 , 0,      0 % 2 , 0,      0,      0 % 2 , 0,      0 % 2 , 0,      0 % 2 , 0 % 2 , 1 % 2 , 0 % 2 , 0,
0 % 2 , 0,      0 % 2 , 0,      0,      0 % 2 , 0,      0 % 2 , 0,      0 % 2 , 0 % 2 , 0,      0 % 2 , 1 %
0 % 2 , 0,      0 % 2 , 0,      0,      0 % 2 , 0,      0 % 2 , 0,      0 % 2 , 0 % 2 , 0,      0 % 2 , 0,
0 % 2 , 0,      0 % 2 , 0,      0,      0 % 2 , 0,      0 % 2 , 0,      0 % 2 , 1 % 2 , 0,      0 % 2 , 0,
0 % 2 , 0,      0 % 2 , 0,      0,      0 % 2 , 0,      0 % 2 , 0,      0 % 2 , 0 % 2 , 0,      1 % 2 , 0,
0 % 2 , 0,      0 % 2 , 0,      0,      0 % 2 , 0,      0 % 2 , 0,      0 % 2 , 0 % 2 , 0,      0 % 2 , 0,
0 % 2 , 0,      0 % 2 , 0,      0,      0 % 2 , 0,      0 % 2 , 0,      0 % 2 , 0 % 2 , 0,      0 % 2 , 0,
0 % 2 , 0,      0 % 2 , 0,      0,      0 % 2 , 0,      0 % 2 , 0,      0 % 2 , 0 % 2 , 0,      0 % 2 , 0,
```

71 COM:=[seq(list2mat(ii,n),ii=com)];

```
[ 0 % 2 , 1 % 2 , 0 % 2 , 0, 0          0 % 2 , 0, 0 % 2 , 1 % 2 , 0          0 % 2 , 0, 0 % 2 , 0, 1 % 2
  0 % 2 , 0,      0 % 2 , 0, 0 % 2      0 % 2 , 0, 0 % 2 , 0,      0 % 2      0 % 2 , 0, 0 % 2 , 0, 0 % 2
  0 % 2 , 0,      0 % 2 , 0, 0          0 % 2 , 0, 0 % 2 , 0,      0          0 % 2 , 0, 0 % 2 , 0, 0
  0 % 2 , 0,      0 % 2 , 0, 0 % 2      0 % 2 , 0, 0 % 2 , 0,      0 % 2      0 % 2 , 0, 0 % 2 , 0, 0 % 2
  0 % 2 , 0,      0 % 2 , 0, 0          0 % 2 , 0, 0 % 2 , 0,      0          0 % 2 , 0, 0 % 2 , 0, 0
```

72 v:=[1$13];

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

73 v*COM;

```
1 % 2 , 1 % 2 , 1 % 2 , 1 % 2 , 1 % 2
0 % 2 , 1 % 2 , 0 % 2 , 1 % 2 , 0 % 2
1 % 2 , 1 % 2 , 1 % 2 , 1 % 2 , 1 % 2
0 % 2 , 1 % 2 , 0 % 2 , 1 % 2 , 0 % 2
0 % 2 , 1 % 2 , 0 % 2 , 1 % 2 , 1 % 2
```

74 [1,2,3]*[a,b,c,d,e];

[[2, 0, a+ 3]]

75 convert(67,base,2);

[1, 1, 0, 0, 0, 0, 1]

76 convert(0,base,2);// Attention, il ne retoure pas 0 mais le vide.

```
77 augment(convert(67,base,2),[0$10]); // pour ajouter des 0
```

$$[1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

```
78 convert(67,base,2)+[0$13]; // la somme est ajustee au vecteur le plus long
```

$$[1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]$$

```
79 Pour acceler la boucle on releve la liste COM dans Z
```

```
80 COM:=COM%0;
```

$$\left[\begin{bmatrix} 0, & 1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \end{bmatrix}, \begin{bmatrix} 0, & 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \end{bmatrix}, \begin{bmatrix} 0, & 0, & 0, & 0, & 1 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \end{bmatrix}, \begin{bmatrix} 1, & 0, & 0, & 0, & 0 \\ 0, & 1, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \end{bmatrix}, \begin{bmatrix} 0, & 0, & 1, & 0, & 0 \\ 0, & 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0 \end{bmatrix}, \begin{bmatrix} 0, \\ 0, \\ 0, \\ 0, \\ 0, \end{bmatrix}$$

```
81 LCOM:=[];
```

$$[]$$

```
82 for(ii:=0;ii<2^13;ii++)
   {
     ci:=convert(ii,base,2);
     LCOM:=[op(LCOM),(ci+[0$13])*COM];
   }
```

Evaluation time: 63.03

Done

```
83 STAB:=select(x->(det(x)%2 <> 0 ),LCOM);
```

// Success

Done

```
84 dim(STAB);
```

1536

85 Le cardinal du stabilisateur est aussi le nombre de bases qui Jordanisent l'endomorphisme J. de partition:

XX qui represente e1e2
XX e3e4
X e5

on choisit d'abord les classes de e2 et e4 dans kerJ^2/kerJ qui est de dim 2, ce qui fait 3*2 choix (le cardinal de GL2(F2)). puis on remonte ces classes dans kerJ^2. (Chaque classe a autant d'antecedants que kerJ:2^3. On a donc 6*8*8 choix pour e2,e4. ils determinent e1 et e3. On choisit e5 dans ker N-(vect(e1,e3)):8-4 choix. Au total: 6*8*8*4=1536

86