1 ---------------------------------EXERCICE-------------------------------------

```
2 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, 0.1000000000

```
3 v1:=[seq(j,j=1..4)];
```

[1, 2, 3, 4]

```
4 v2:=[a,b,c,d];
```

[a, b, c, d]

```
5 M:=[[1,2,3,4],[5,6,7,8]];
```

$\begin{bmatrix}1 & 2 & 3 & 4\\5 & 6 & 7 & 8\end{bmatrix}$

```
6 M[0];
```

[1, 2, 3, 4]

```
7 M
```

$\begin{bmatrix}1 & 2 & 3 & 4\\5 & 6 & 7 & 8\end{bmatrix}$

```
8 [v1]; // une matrice ligne
```

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

```
9 M*v1; //est bien un vecteur.
```

[30, 70]

```
10 diag(seq(1,4)); diag(1$4);
```

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

```
11 diag(seq(j,j=1..4));
```

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

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

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

```
13 [op(A),op(v)]; //ajoute une ligne facilement.
```

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

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

15 V:=matrix(6,6,(k,l)->x[k]^l);

// Attention: x, declaree(s) comme variable(s) globale(s)

$$\begin{bmatrix} 1, & x[0], & (x[0])^2, & (x[0])^3, & (x[0])^4, & (x[0])^5 \\ 1, & x[1], & (x[1])^2, & (x[1])^3, & (x[1])^4, & (x[1])^5 \\ 1, & x[2], & (x[2])^2, & (x[2])^3, & (x[2])^4, & (x[2])^5 \\ 1, & x[3], & (x[3])^2, & (x[3])^3, & (x[3])^4, & (x[3])^5 \\ 1, & x[4], & (x[4])^2, & (x[4])^3, & (x[4])^4, & (x[4])^5 \\ 1, & x[5], & (x[5])^2, & (x[5])^3, & (x[5])^4, & (x[5])^5 \end{bmatrix}$$

M

16 dv:=det(V);

Evaluation time: 3.64

Done

17 factor(dv);

(-(x[4])+ x[5])*(x[3] -(x[4]))*(x[3] -(x[5]))*(x[2] -(x[3]))*(x[2] -(x[5]))*(x[2] -(x[4]))*(x[1] -(x[2]))*(x[1] -(x[4]

M

18 purge(a);

No such variable a

M

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

// Attention: a, declaree(s) comme variable(s) globale(s)
// End defining f

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

M

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

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

M

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

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

M

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

$$\begin{bmatrix} 0, & a[0,1], & a[0,2], & a[0,3], & a[0,4], & a[0,5], & a[0,6], & a \\ -(a[0,1]), & 0, & a[1,2], & a[1,3], & a[1,4], & a[1,5], & a[1,6], & a \\ -(a[0,2]), & -(a[1,2]), & 0, & a[2,3], & a[2,4], & a[2,5], & a[2,6], & a \\ -(a[0,3]), & -(a[1,3]), & -(a[2,3]), & 0, & a[3,4], & a[3,5], & a[3,6], & a \\ -(a[0,4]), & -(a[1,4]), & -(a[2,4]), & -(a[3,4]), & 0, & a[4,5], & a[4,6], & a \\ -(a[0,5]), & -(a[1,5]), & -(a[2,5]), & -(a[3,5]), & -(a[4,5]), & 0, & a[5,6], & a \\ -(a[0,6]), & -(a[1,6]), & -(a[2,6]), & -(a[3,6]), & -(a[4,6]), & -(a[5,6]), & 0, & a \\ -(a[0,7]), & -(a[1,7]), & -(a[2,7]), & -(a[3,7]), & -(a[4,7]), & -(a[5,7]), & -(a[6,7]), & 0 \end{bmatrix}$$

M

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

24 d1:=det(M):;

Evaluation time: 5.52

Done

M

25 d2:=det_minor(M):;

Done

26 normal(d1-d2);

Evaluation time: 1.94

$$0$$

27 --------------------------------EXERCICE-------------------------------------

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

// Success
// End defining fjj

(ii, j)-> if ((ii==(j-1))) 1;

29 JJ(n):=matrix(n,n,fjj);//forme classique d'ordre 8.

// Interprete JJ
// Attention: fjj, declaree(s) comme variable(s) globale(s) compiling JJ

n -> matrix(n ,n ,fjj)

30 J(L):=diag([seq(JJ(l),l=L)]);//l parcourt la liste L

// Interprete J
// Attention: l, declaree(s) comme variable(s) globale(s) compiling J

L -> diag([seq(JJ(l) ,l= L)])

31 N:=[[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

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

32 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.

33 N,N^2;

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

34 N2:=nullspace(N^2);

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

35 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).

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

(Done , Done)

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

8

38 N1:=nullspace(N);

-1	0	0	0	0	0	0	0
0	0	0	-1	0	0	0	0
0	0	0	0	0	-2	-1	0

39 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:

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

Done

41 On verifie qu'il convient:

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

6

43 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)

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

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

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

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

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

47 --

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

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

50
```
f:=(ii,j)->if(ii=j) then 1 else 0 fi;
```

```
// Success
// End defining f
```

(ii, j)-> if ((ii==j)) 1;

M

51
```
T:=proc(ii,j,a)
local A;
A:=matrix(8,8,f)::A[ii-1,j-1]:=a;A;
end proc;
```

```
// Attention: f, declaree(s) comme variable(s) globale(s)
// End defining T
```

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

M

52
```
B:=matrix(8,8,b);
```

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,	0,	0,	0

M

53 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);

54
```
purge(a);
```

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

M

55
```
T(3,2,a)*B;
```

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,	0,	0,	0

M

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

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

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

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

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

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

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

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

Done

62 P,P^(-1);

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

63 Donc faire a l'ordinateur:

64 N:=P*J([3,3,2])*P^(-1)

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

65 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