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

2 gcd(6,3);#Attention, a priori pour lui c'est des polynomes, on a dela chance il les normalise bien.

3

3 gcd(x*(x+3),2*x);

x

4 igcd(4,6,8);

2

5 iquo(13,6);igcd(6,0);

(2, 6)

6 igcdex(4,15,'a','b');#pour un couple de bezout:

1

7 4*a+15*b;

1

8 delrows(matrix([[1,1],[2,3]]),2..2); #pour une colonne: delcols

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

9 A:=diag([3,6,18,36]);B:=matrix(4,4,(i,j)->rand(7)-3);det(B);

// Success

$$\left(\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 18 & 0 \\ 0 & 0 & 0 & 36 \end{bmatrix}, \begin{bmatrix} 3 & 2 & -1 & -2 \\ 0 & 2 & 1 & 3 \\ 1 & 2 & -3 & -1 \\ 2 & 1 & 1 & -3 \end{bmatrix}, 49 \right)$$

10 ismith(A*B);

$$\left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 10 & -28 & 1 & 7 \\ -6444 & -987 & 717 & 703 \\ 18588 & 2844 & -2068 & -2027 \end{bmatrix}, \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 18 & 0 \\ 0 & 0 & 0 & 1764 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 34 \\ 0 & -111 & 9315 & 316487 \\ -1 & -220 & 18465 & 627368 \\ 0 & -1 & 84 & 2854 \end{bmatrix} \right)$$

11 --

12 Prog Edit Add nxt OK Save

```
minval:= proc(A)
local m,u,v,i,j;
m:=0;u:=0;v:=0;
# NB: On retourne 0,0  si A est nulle
for i from 1 to dim(A)[1]  do
for j from 1 to dim(A)[2]  do
if ((abs(A[i,j])>0)  and ((m=0)  or (abs(A[i,j])<m)))  then m:=abs(A[i,j]);u:=i;v:
od;
od;
u,v;end  proc;
```

```
proc(A)
  local m,u,v,i,j;
  m:=0;
  u:=0;
  v:=0;
  for i from 1 to (dim(A))[1]+1/2 do
  for j from 1 to (dim(A))[2]+1/2 do
  if ((abs(A[i,j]))>0) and ((m=0) or ((abs(A[i,j]))<m)) then
    m:=abs(A[i,j]);
    u:=i;
    v:=j
  fi ;
   od;;
   od;;
  u,v;
```

13 A:=matrix([[6, 0, 7, 9], [6, 8, 6, 9], [3, 2, 4, 6], [3, 2, 3, 3]]);

$$\begin{bmatrix} 6 & 0 & 7 & 9 \\ 6 & 8 & 6 & 9 \\ 3 & 2 & 4 & 6 \\ 3 & 2 & 3 & 3 \end{bmatrix}$$

Menu

14 minval(A);

(3, 2)

Menu

15 (i,j):=minval(A);

(3, 2)

Menu

16 U:=identity(4);

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

Menu

17 normalement il faudrait faire bouger l dans {1..4} sauf j, c'est plus simple de
ne pas mettre de test, et de corriger U[j,j] ensuite

18

undef

Menu

19 for l from 1 to 4 do U[j,l]:=-iquo(A[i,l],A[i,j]) od:U[j,j]:=1:U;

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

Menu

20 A*U;

$$\begin{bmatrix} 6 & 0 & 7 & 9 \\ -2 & 8 & -10 & -15 \\ 1 & 2 & 0 & 0 \\ 1 & 2 & -1 & -3 \end{bmatrix}$$

Menu

21

22 Prog Edit Add | nxt | OK | Save

```
trans:= proc (A,i,j)
local n,U,V,l;
n:=dim(A)[1];
U:=identity(n);
V:=identity(n);
for l from 1 to n do
V[l,i]:=-iquo(A[l,j],A[i,j]);
od;
//on corrige
V[i,i]:=1;
for l from 1 to n do
U[j,l]:=-iquo(A[i,l],A[i,j]);
od;
//on corrige
U[j,j]:=1;
V*A*U;
end proc;
```

// End defining trans

```
proc(A,i,j)
  local n,U,V,l;
  n:=(dim(A))[1];
  U:=identity(n);
  V:=identity(n);
  for l from 1 to n+1/2 do
  V[l,i]:=-(iquo(A[l,j],A[i,j]));
   od;;
  V[i,i]:=1;
  for l from 1 to n+1/2 do
```

```
od;;
U[j,j]:=1;
V*A*U;

end;
```

Menu

23 A:=matrix([[6, 1, 7, 9], [6, 8, 6, 9], [3, 1, 4, 6], [3, 2, 1, 3]]);

$$\begin{bmatrix} 6 & 1 & 7 & 9 \\ 6 & 8 & 6 & 9 \\ 3 & 1 & 4 & 6 \\ 3 & 2 & 1 & 3 \end{bmatrix}$$

Menu

24 trans(A,3,1);

$$\begin{bmatrix} 0 & -1 & -1 & -3 \\ 0 & 6 & -2 & -3 \\ 3 & 1 & 1 & 0 \\ 0 & 1 & -3 & -3 \end{bmatrix}$$

Menu

25

26 Prog Edit Add nxt OK Save

```
Zequiv:= proc(A)
k:=0;
n:=dim(A)[1];
l:=[seq(0,k=1..n)];
B:=A;
(i,j):=minval(B);
//minval retourne les coordonnees du minimum non nul,
//ou un couple impossible (Ex ici (0,0) en mode maple) si B est nulle.
while([i,j]<>[0,0]) and size(B)>1 do
// attention, ne pas oublier de declarer les variables locales
// dans trans pour  eviter les melanges
B:=trans(B,i,j);
if [i,j]=[minval(B)]  then
// on n'a que des zeros sur la ligne i et la colonne j sauf
// en (i,j) on sauve donc B[i,j] et on raye ligne et colonne.
k:=k+1; l[k]:=B[i,j];
B:=delcols(B,j..j);
B:=delrows(B,i..i);
//le cas B de taille 1 pose probleme car on ne peut pas rayer
//une colonne puis une ligne
fi;
// ASTUCE: voici comment assigner 2 valeurs d'un coup.
(i,j):=minval(B);
end do;
// On a eventuellement change le signe du determinant
diag([seq(l[k],k=1..n-1),B[1,1]]);
end proc;
```

// End defining Zequiv

```
proc(A)
 k:=0;
 n:=(dim(A))[1];
 l:=[seq(0,k=(1 .. n))];
 B:=A;
 i,j:=minval(B);
 while (([i,j])<>[0,0]) and ((nops(B))>1) do
   B:=trans(B,i,j);
   if [i,j]=[minval(B)] then
     k:=k+1;
     l[k]:=B[i,j];
     B:=delcols(B,j .. j);
     B:=delrows(B,i .. i)
   fi ;
   i,j:=minval(B);
  od;;
 diag(seq(l[k],k=(1 .. (n-1))),B[1,1]);
```

27 A:=matrix([[2,2,2],[6,12,6],[6,4,6]]); #Exemple:

$$\begin{bmatrix} 2 & 2 & 2 \\ 6 & 12 & 6 \\ 6 & 4 & 6 \end{bmatrix}$$

Menu

28 Zequiv(A);

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

Menu

29 A[3,3]:=12:A;

$$\left(\text{Done}, \begin{bmatrix} 2 & 2 & 2 \\ 6 & 12 & 6 \\ 6 & 4 & 12 \end{bmatrix} \right)$$

Menu

30 Zequiv(A);

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 18 \end{bmatrix}$$

Menu

31 On v'erifie que A et Zequiv(A) ont bien meme forme de smith:

32 ismith(A)[2], ismith(Zequiv(A))[2];

$$\left(\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 18 \end{bmatrix}, \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 18 \end{bmatrix} \right)$$

Menu

33 Attention, on n'obtient pas forcement les diviseurs elementaires, par exemple:

34 A:=matrix([[4,0,0],[0,6,0],[0,0,8]]);

$$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 8 \end{bmatrix}$$

Menu

35 ismith(A)[2];

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 24 \end{bmatrix}$$

Menu

36 Zequiv(A);

$$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 8 \end{bmatrix}$$

Menu

37 ca sera forc'ement le pgcd(d_1,...,d_n) car l'algorithme reste sur la premiere ligne.

38 f:=(i,j)->if (i-j)*(i-1)=0 then 1 else 0 fi;

// Success
// End defining f

```
          if (i-j)*(i-1)=0 then
            1 else
            0
( i,  j  )->  fi
```

Menu

39 matrix(3,3,f);A:=matrix(3,3,f)*A;Zequiv(A);

$$\left(\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 4 & 6 & 8 \\ 0 & 6 & 0 \\ 0 & 0 & 8 \end{bmatrix}, \begin{bmatrix} 2 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & -12 \end{bmatrix} \right)$$

Menu

40 On recopie trans et Zequiv pour qu'ils ne travaillent que sur les colonnes

41 Prog Edit Add | nxt | OK | Save

```
transC:= proc(A,i,j)
//attention, si on ne met pas un local n, il modifiera la valeur de n dans Zequ
local n,U,l;
n:=dim(A)[1];
U:=identity(n);
for l from 1 to n do
U[j,l]:=-iquo(A[i,l],A[i,j]);
od;
//pour ne pas mettre de test dans la boucle precedente, on n'ecarte pas l=j, ma
U[j,j]:=1;
A*U;
end proc;
```

```
proc(A,i,j)
  local n,U,l;
  n:=(dim(A))[1];
  U:=identity(n);
  for l from 1 to n+1/2 do
  U[j,l]:=-(iquo(A[i,l],A[i,j]));
   od;;
  U[j,j]:=1;
  A*U;
```

Menu

42

43 Prog Edit Add | nxt | OK | Save

```
ZequivC:= proc(A)
k:=0;
n:=dim(A)[1];
l:=[seq(0,k=1..n)];
B:=A;
(i,j):=minval(B);
while([i,j]<>[0,0]) and size(B)>1 do
// attention, ne pas oublier de declarer les variables locales dans trans pour
B:=transC(B,i,j);
if [i,j]=[minval(B)] then
k:=k+1; l[k]:=B[i,j];
B:=delcols(B,j..j);
B:=delrows(B,i..i);
fi;
// ASTUCE: voici comment assigner 2 valeurs d'un coup.
(i,j):=minval(B);
end do;
// On a eventuellement change le signe du determinant
diag([seq(l[k],k=1..n-1),B[1,1]]);
end proc;
```

```
proc(A)
  k:=0;
  n:=(dim(A))[1];
  l:=[seq(0,k=(1 .. n))];
  B:=A;
  i,j:=minval(B);
  while (([i,j])<>[0,0]) and ((nops(B))>1) do
    B:=transC(B,i,j);
    if [i,j]=[minval(B)] then
      k:=k+1;
      l[k]:=B[i,j];
      B:=delcols(B,j .. j);
      B:=delrows(B,i .. i)
    fi ;
    i,j:=minval(B);
   od;;
  diag(seq(l[k],k=(1 .. (n-1))),B[1,1]);
```

Menu

44

45 Prog Edit Add nxt OK Save

```
elem:= proc (A)
n:=dim(A)[1];
d:=Zequiv(A);
L:=[];
for i from 1 to n-1 do
T:=matrix(n+1-i,n+1-i,f);
d:=ZequivC(T*d);
L:=[op(L),d[1,1]];
d:=delrows(delcols(d,1..1),1..1);
od;
[op(L),d[1,1]];
end proc;
```

// Warning: n Zequiv d L i f T ZequivC declared as global variable(s)
// End defining elem

proc(A)
n:=(dim(A))[1];

Menu

46 A:=matrix([[2,2,2],[6,12,6],[6,4,12]]);

$$\begin{bmatrix} 2 & 2 & 2 \\ 6 & 12 & 6 \\ 6 & 4 & 12 \end{bmatrix}$$

Menu

47 elem(A);ismith(A)[2];

$$\left([2 \quad -2 \quad 18], \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 18 \end{bmatrix}\right)$$

Menu

48 elem(diag(4,6,16));ismith(diag(4,6,16))[2];

$$\left([2 \quad 4 \quad 48], \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 48 \end{bmatrix}\right)$$

Menu

49