```
    |\^/|     Maple 9 (IBM INTEL LINUX)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2003
 \  MAPLE  /  All rights reserved. Maple is a trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
>  interface(screenwidth=120);
> with(linalg):with(LinearAlgebra):
> gcd(6,3);#Attention, a priori pour lui c'est des polynomes, on a dela chance il les normalise bien.
                                                             3

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

> igcd(4,6,8);
                                                             2

> iquo(13,6);igcd(6,0);
                                                             2

                                                             6

>  igcdex(4,15,'a','b');#pour un couple de bezout:
                                                             1

>  4*a+15*b;
                                                             1

> DeleteRow(Matrix([[1,1],[2,3]]),2);#pour une colonne: DeleteColumn
                                                          [1    1]

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

>  #############
>  trans:=proc(A,i,j)
>  n:=Dimension(A)[1];
>  U:=Matrix(n,n)+1;
>  V:=Matrix(n,n)+1;
>  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;
trans := proc(A, i, j)
local n, U, V, l;
    n := LinearAlgebra:-Dimension(A)[1];
    U := Matrix(n, n) + 1;
    V := Matrix(n, n) + 1;
    for l to n do V[l, i] := -iquo(A[l, j], A[i, j]) end do;
    V[i, i] := 1;
    for l to n do U[j, l] := -iquo(A[i, l], A[i, j]) end do;
    U[j, j] := 1;
    V . A . U
end proc

>  #####
>  A:=Matrix([[6, 1, 7, 9], [6, 8, 6, 9], [3, 1, 4, 6], [3, 2, 1, 3]]);
                                                      [6    1    7    9]
                                                      [                ]
                                                      [6    8    6    9]
                                                 A := [                ]
                                                      [3    1    4    6]
                                                      [                ]
                                                      [3    2    1    3]

>  trans(A,3,1);
                                                  [0    -1    -1    -3]
                                                  [                   ]
                                                  [0     6    -2    -3]
                                                  [                   ]
                                                  [3     1     1     0]
                                                  [                   ]
                                                  [0     1    -3    -3]

>  Zequiv:=proc(A)
>  k:=0;
>  n:=Dimension(A)[1];
>  l:=[seq(0,k=1..n)];
>  B:=A;
>  (i,j):=minval(B);
>  while((i,j)<>(0,0)) do
>  B:=trans(B,i,j);
>  if (i,j)=minval(B) then k:=k+1;l[k]:=B[i,j];B:=DeleteRow(DeleteColumn(B,j),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
>  DiagonalMatrix([seq(l[k],k=1..n)]);
>  end proc;
Zequiv := proc(A)
local k, n, l, B, i, j;
    k := 0;
    n := LinearAlgebra:-Dimension(A)[1];
    l := [seq(0, k = 1 .. n)];
    B := A;
    i, j := minval(B);
    while (i, j) <> (0, 0) do
        B := trans(B, i, j);
        if (i, j) = minval(B) then
            k := k + 1; l[k] := B[i, j]; B := LinearAlgebra:-DeleteRow(LinearAlgebra:-DeleteColumn(B, j), i)
        end if;
        i, j := minval(B)
    end do;
    LinearAlgebra:-DiagonalMatrix([seq(l[k], k = 1 .. n)])
end proc

>  #Exemple:
>  A:=Matrix([[2,2,2],[6,12,6],[6,4,6]]);
                                                      [2     2    2]
                                                      [            ]
                                                 A := [6    12    6]
                                                      [            ]
                                                      [6     4    6]

>  Zequiv(A);
                                                    [2     0    0]
                                                    [            ]
                                                    [0    -2    0]
                                                    [            ]
                                                    [0     0    0]

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

>  Zequiv(A);
                                                   [2     0     0]
                                                   [             ]
                                                   [0    -2     0]
                                                   [             ]
                                                   [0     0    18]

>  #On v\'erifie que A et Zequiv(A) ont bien meme forme de smith:
>  ismith(A), ismith(Zequiv(A));
                                           [2    0     0]  [2    0     0]
                                           [            ]  [            ]
                                           [0    2     0], [0    2     0]
                                           [            ]  [            ]
                                           [0    0    18]  [0    0    18]

#Non, par exemple:
> A:=Matrix([[4,0,0],[0,6,0],[0,0,8]]);
                                                      [4    0    0]
                                                      [           ]
                                                 A := [0    6    0]
                                                      [           ]
                                                      [0    0    8]

#ca sera forc\'ement le pgcd(d_1,...,d_n) car  l'algorithme reste sur la premiere ligne.
> f:=(i,j)->if (i-j)*(i-1)=0 then 1 else 0 fi;
              f := proc(i, j) option operator, arrow; if (i - j)*(i - 1) = 0 then 1 else 0 end if end proc

> Matrix(3,3,f);A:=Matrix(3,3,f).A;Zequiv(A);
                                                   [1    1    1]
                                                   [           ]
                                                   [0    1    0]
                                                   [           ]
                                                   [0    0    1]

                                                      [4    6    8]
                                                      [           ]
                                                 A := [0    6    0]
                                                      [           ]
                                                      [0    0    8]

bytes used=4000208, alloc=3341724, time=0.14
                                                   [2    0      0]
                                                   [             ]
                                                   [0    8      0]
                                                   [             ]
                                                   [0    0    -12]

>  transC:=proc(A,i,j)
>  n:=Dimension(A)[1];
>  U:=Matrix(n,n)+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;
>  A.U;
>  end proc;
transC := proc(A, i, j)
local n, U, l;
    n := LinearAlgebra:-Dimension(A)[1];
    U := Matrix(n, n) + 1;
    for l to n do U[j, l] := -iquo(A[i, l], A[i, j]) end do;
    U[j, j] := 1;
    A . U
end proc

>  ZequivC:=proc(A)
>  k:=0;
>  n:=Dimension(A)[1];
>  l:=[seq(0,k=1..n)];
>  B:=A;
>  (i,j):=minval(B);
>  while((i,j)<>(0,0)) do
>  B:=transC(B,i,j);
>  if (i,j)=minval(B) then k:=k+1;l[k]:=B[i,j];B:=DeleteRow(DeleteColumn(B,j),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
>  DiagonalMatrix([seq(l[k],k=1..n)]);
>  end proc;
ZequivC := proc(A)
local k, n, l, B, i, j;
    k := 0;
    n := LinearAlgebra:-Dimension(A)[1];
    l := [seq(0, k = 1 .. n)];
    B := A;
    i, j := minval(B);
    while (i, j) <> (0, 0) do
        B := transC(B, i, j);
        if (i, j) = minval(B) then
            k := k + 1; l[k] := B[i, j]; B := LinearAlgebra:-DeleteRow(LinearAlgebra:-DeleteColumn(B, j), i)
        end if;
        i, j := minval(B)
    end do;
    LinearAlgebra:-DiagonalMatrix([seq(l[k], k = 1 .. n)])
end proc

>  elem:=proc(A)
>  n:=Dimension(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:=DeleteRow(DeleteColumn(D,1),1);
>  od;
>  [op(L),D[1,1]];
>  end proc;
elem := proc(A)
local n, D, L, i, T;
    n := LinearAlgebra:-Dimension(A)[1];
    D := Zequiv(A);
    L := [];
    for i to n - 1 do
        T := Matrix(n - i + 1, n - i + 1, f);
        D := ZequivC(T . D);
        L := [op(L), D[1, 1]];
        D := LinearAlgebra:-DeleteRow(LinearAlgebra:-DeleteColumn(D, 1), 1)
    end do;
    [op(L), D[1, 1]]
end proc

>  A:=Matrix([[2,2,2],[6,12,6],[6,4,12]]);
                                                      [2     2     2]
                                                      [             ]
                                                 A := [6    12     6]
                                                      [             ]
                                                      [6     4    12]

>  elem(A);
                                                     [2, -2, 18]

> quit
bytes used=5551920, alloc=3341724, time=0.18
```