```
> restart;
> with(linalg);
```

[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]

Fonctions de base

```
> matrice:=proc(E)local n,m,d,s,i,j,k;n:=rowdim(E);d:=NULL;for i
  from 1 to n-1 do    s:=0;    for j from 1 to n do
  s:=s+E[i,j];    od;
  d:=d,s;od;m:=matrix(n-1,n-1,(i,j)->-E[i,j]);for k from 1 to n-1
  do    m[k,k]:=d[k]+m[k,k];od;[m,[d]];end;
```

```
matrice := proc(E)
local n, m, d, s, i, j, k;
    n := linalg:-rowdim(E);
    d := NULL;
    for i to n − 1 do s := 0; for j to n do s := s + E[i, j] end do; d := d, s end do;
    m := linalg:-matrix(n − 1, n − 1, (i, j) → −E[i, j]);
    for k to n − 1 do m[k, k] := d[k] + m[k, k] end do;
    [m, [d]]
end proc
```

```
> eboulement:=proc(u,m,i)
  local j,n,v;
  n:=nops(u);
  v:=NULL;
  for j from 1 to n do v:=v,u[j]-m[i,j] od;
  [v];
  end;
```

```
eboulement := proc(u, m, i)
local j, n, v;
    n := nops(u); v := NULL; for j to n do v := v, u[j] − m[i, j] end do; [v]
end proc
```

```
> avalanche:=proc(u,m)
  local n,i,v;
  n:=nops(u);
  v:=u;
```

```
  i:=1;
  while i<n+1 do
        if v[i]<m[i,i] then i:=i+1
        else v:=eboulement(v,m,i);
             i:=1
        fi;
  od;
  v;
  end;
```

avalanche := proc(u, m)
local n, i, v;
    n := nops(u);
    v := u;
    i := 1;
    while i < n + 1 do if v[i] < m[i, i] then i := i + 1 else v := eboulement(v, m, i); i := 1 end if
    end do;
    v
end proc

```
> beta:=proc(m,n)
  local l,i,s,j;
  l:=NULL;
  for i from 1 to n do
      s:=0;
      for j from 1 to n do s:=s+m[i,j] od;
      l:=l,s
  od;
  [l]
  end;
```

β := proc(m, n)
local l, i, s, j;
    l := NULL; for i to n do s := 0; for j to n do s := s + m[i, j] end do; l := l, s end do; [l]
end proc

```
> successeur:=proc(u,m,d,n)
  local v,i;
  i:=n;
  v:=u;
  while i<>0 and u[i]=d[i]-1 do
        v[i]:=0;
        i:=i-1;
  od;
  if i=0 then 0 else v[i]:=v[i]+1;
                     v
  fi;
  end;
```

successeur := proc(u, m, d, n)
local v, i;
    i := n;

$v := u;$
while $i \neq 0$ and $u[i] = d[i] - 1$ do $v[i] := 0$; $i := i - 1$ end do;
if $i = 0$ then 0 else $v[i] := v[i] + 1$; v end if
end proc

```
> recurrentes:=proc(m,d)
  local u,n,b,l;
  l:=NULL;
  n:=rowdim(m);
  u:=[seq(0,i=1..n)];
  b:=beta(m,n);
  while u<>0 do
      while u<>0 and avalanche(u+b,m)<>u do
            u:=successeur(u,m,d,n);
      od;
      if u<>0 then l:=l,u;
                   u:=successeur(u,m,d,n);
      fi;
  od;
  [l];
  end;
```

recurrentes := proc(m, d)
local u, n, b, l;
$l :=$ NULL;
$n :=$ linalg:-rowdim(m);
$u := [\mathrm{seq}(0, i = 1 .. n)]$;
$b := \beta(m, n)$;
while $u \neq 0$ do
while $u \neq 0$ and avalanche$(u + b, m) \neq u$ do $u :=$ successeur(u, m, d, n) end do;
if $u \neq 0$ then $l := l, u$; $u :=$ successeur(u, m, d, n) end if
end do;
[l]
end proc

```
> delta:=proc(m)
  local n;
  n:=rowdim(m);
  [seq('m[i,i]',i=1..n)];
  end;
```

$\delta :=$ proc(m) local n; $n :=$ linalg:-rowdim(m); [seq('m[i, i]', $i = 1 .. n$)] end proc

```
> epsilon:=proc(m)
  local d;
  d:=delta(m):
  d+(d-avalanche(d+d,m));
  end;
```

$\varepsilon :=$ proc(m) local d; $d := \delta(m)$; $2*d -$ avalanche$(2*d, m)$ end proc

La maison

```
> E:=matrix(5,5,[[
  0,1,1,0,0],[1,0,1,1,0],[1,1,0,2,1],[0,1,2,0,1],[0,0,1,1,0]]);
```

$$E := \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 2 & 1 \\ 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

```
> mat:=matrice(E);
```

$$mat := [m, [2, 3, 5, 4]]$$

```
> m:=mat[1];d:=mat[2];
```

$$m := m$$
$$d := [2, 3, 5, 4]$$

```
> evalm(m);
```

$$\begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 5 & -2 \\ 0 & -1 & -2 & 4 \end{bmatrix}$$

```
> eboulement([2,3,3,1],m,1);
```

$$[0, 4, 4, 1]$$

```
> avalanche([2,3,3,1],m);
```

$$[1, 1, 4, 1]$$

```
> beta(m,4);
```

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

```
> successeur([1,2,4,3],m,d,4);
```

$$0$$

```
> recurrentes(m,d);
```

[[0, 1, 2, 3], [0, 1, 3, 3], [0, 1, 4, 1], [0, 1, 4, 2], [0, 1, 4, 3], [0, 2, 1, 3], [0, 2, 2, 3], [0, 2, 3, 3], [0, 2, 4, 0], [0, 2, 4, 1], [0, 2, 4, 2], [0, 2, 4, 3], [1, 0, 2, 3], [1, 0, 3, 3], [1, 0, 4, 1], [1, 0, 4, 2], [1, 0, 4, 3], [1, 1, 2, 3], [1, 1, 3, 3], [1, 1, 4, 0], [1, 1, 4, 1], [1, 1, 4, 2], [1, 1, 4, 3], [1, 2, 0, 3], [1, 2, 1, 3], [1, 2, 2, 3], [1, 2, 3, 3], [1, 2, 4, 0], [1, 2, 4, 1], [1, 2, 4, 2], [1, 2, 4, 3]]

```
> nops(%);
```

$$31$$

```
>
> s:=ismith(m,u,v);
```

$$s := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 31 \end{bmatrix}$$

```
> eval(u);
```

$$\begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 14 & 15 & 13 & 18 \end{bmatrix}$$

```
> eval(v);
```

$$\begin{bmatrix} 1 & -1 & 8 & -13 \\ 0 & 1 & -3 & 6 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> evalm(u &* m &* v - s);
```

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

```
> evalm(u^(-1));
```

$$\begin{bmatrix} 2 & -3 & 18 & -1 \\ -1 & 4 & -18 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \end{bmatrix}$$

```
> delta(m);
```

$$[2, 3, 5, 4]$$

```
> epsilon(m);
```

$$[4, 4, 8, 5]$$

```
> avalanche([38,2,40,26],m);
```

$$[1, 2, 2, 3]$$

```
> eps:=avalanche(epsilon(m),m);
```

$$eps := [1, 2, 2, 3]$$

```
> avalanche(eps+[-1,1,0,0],m);
```

$$[1, 0, 2, 3]$$

```
>
>
```

Roue

```
> m:=<<3,-1,0,0,-1>|<-1,3,-1,0,0>|<0,-1,3,-1,0>|<0,0,-1,3,-1>|<-1,
  0,0,-1,3>>;
```

$$m := \begin{bmatrix} 3 & -1 & 0 & 0 & -1 \\ -1 & 3 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 3 \end{bmatrix}$$

```
> whattype(m);
```

Matrix

```
> s:=linalg[ismith](m,u,v);
```

$$s := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 11 & 0 \\ 0 & 0 & 0 & 0 & 11 \end{bmatrix}$$

```
> recurrentes(m,[3,3,3,3,3]);
```

[[0, 1, 1, 1, 2], [0, 1, 1, 2, 1], [0, 1, 1, 2, 2], [0, 1, 2, 0, 2], [0, 1, 2, 1, 1], [0, 1, 2, 1, 2],
[0, 1, 2, 2, 1], [0, 1, 2, 2, 2], [0, 2, 0, 1, 2], [0, 2, 0, 2, 1], [0, 2, 0, 2, 2], [0, 2, 1, 0, 2],
[0, 2, 1, 1, 1], [0, 2, 1, 1, 2], [0, 2, 1, 2, 1], [0, 2, 1, 2, 2], [0, 2, 2, 0, 2], [0, 2, 2, 1, 1],
[0, 2, 2, 1, 2], [0, 2, 2, 2, 1], [0, 2, 2, 2, 2], [1, 0, 1, 1, 2], [1, 0, 1, 2, 1], [1, 0, 1, 2, 2],
[1, 0, 2, 0, 2], [1, 0, 2, 1, 1], [1, 0, 2, 1, 2], [1, 0, 2, 2, 1], [1, 0, 2, 2, 2], [1, 1, 0, 1, 2],
[1, 1, 0, 2, 1], [1, 1, 0, 2, 2], [1, 1, 1, 0, 2], [1, 1, 1, 1, 2], [1, 1, 1, 2, 0], [1, 1, 1, 2, 1],
[1, 1, 1, 2, 2], [1, 1, 2, 0, 1], [1, 1, 2, 0, 2], [1, 1, 2, 1, 0], [1, 1, 2, 1, 1], [1, 1, 2, 1, 2],
[1, 1, 2, 2, 0], [1, 1, 2, 2, 1], [1, 1, 2, 2, 2], [1, 2, 0, 1, 1], [1, 2, 0, 1, 2], [1, 2, 0, 2, 0],
[1, 2, 0, 2, 1], [1, 2, 0, 2, 2], [1, 2, 1, 0, 1], [1, 2, 1, 0, 2], [1, 2, 1, 1, 0], [1, 2, 1, 1, 1],
[1, 2, 1, 1, 2], [1, 2, 1, 2, 0], [1, 2, 1, 2, 1], [1, 2, 1, 2, 2], [1, 2, 2, 0, 1], [1, 2, 2, 0, 2],
[1, 2, 2, 1, 0], [1, 2, 2, 1, 1], [1, 2, 2, 1, 2], [1, 2, 2, 2, 0], [1, 2, 2, 2, 1], [1, 2, 2, 2, 2],
[2, 0, 1, 1, 1], [2, 0, 1, 1, 2], [2, 0, 1, 2, 0], [2, 0, 1, 2, 1], [2, 0, 1, 2, 2], [2, 0, 2, 0, 1],
[2, 0, 2, 0, 2], [2, 0, 2, 1, 0], [2, 0, 2, 1, 1], [2, 0, 2, 1, 2], [2, 0, 2, 2, 0], [2, 0, 2, 2, 1],
[2, 0, 2, 2, 2], [2, 1, 0, 1, 1], [2, 1, 0, 1, 2], [2, 1, 0, 2, 0], [2, 1, 0, 2, 1], [2, 1, 0, 2, 2],
[2, 1, 1, 0, 1], [2, 1, 1, 0, 2], [2, 1, 1, 1, 0], [2, 1, 1, 1, 1], [2, 1, 1, 1, 2], [2, 1, 1, 2, 0],
[2, 1, 1, 2, 1], [2, 1, 1, 2, 2], [2, 1, 2, 0, 1], [2, 1, 2, 0, 2], [2, 1, 2, 1, 0], [2, 1, 2, 1, 1],
[2, 1, 2, 1, 2], [2, 1, 2, 2, 0], [2, 1, 2, 2, 1], [2, 1, 2, 2, 2], [2, 2, 0, 1, 1], [2, 2, 0, 1, 2],
[2, 2, 0, 2, 0], [2, 2, 0, 2, 1], [2, 2, 0, 2, 2], [2, 2, 1, 0, 1], [2, 2, 1, 0, 2], [2, 2, 1, 1, 0],
[2, 2, 1, 1, 1], [2, 2, 1, 1, 2], [2, 2, 1, 2, 0], [2, 2, 1, 2, 1], [2, 2, 1, 2, 2], [2, 2, 2, 0, 1],
[2, 2, 2, 0, 2], [2, 2, 2, 1, 0], [2, 2, 2, 1, 1], [2, 2, 2, 1, 2], [2, 2, 2, 2, 0], [2, 2, 2, 2, 1],
[2, 2, 2, 2, 2]]

```
> nops(%);
```

121

```
> evalm(u^(-1));
```

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

```
> epsilon(m);
```

[4, 4, 4, 4, 4]

```
> avalanche(%,m);
```

[2, 2, 2, 2, 2]

```
> u1:=avalanche([4,7,3,4,4],m);
```

u1 := [2, 1, 1, 1, 1]

```
> u2:=avalanche([6,4,7,6,6],m);
```

u2 := [0, 2, 1, 1, 1]

```
> v:=u1;for i from 2 to 11 do v:=avalanche(u1+v,m) od;v;
```

v := [2, 1, 1, 1, 1]
v := [0, 2, 1, 1, 2]
v := [1, 2, 1, 1, 2]
v := [2, 2, 1, 1, 2]
v := [2, 0, 2, 2, 0]
v := [0, 1, 2, 2, 1]
v := [1, 1, 2, 2, 1]
v := [2, 1, 2, 2, 1]
v := [0, 2, 2, 2, 2]
v := [1, 2, 2, 2, 2]
v := [2, 2, 2, 2, 2]
[2, 2, 2, 2, 2]

```
> roue:=proc(n)
  local m,i,j;
  m:=array(1..n,1..n);
  for i from 1 to n do
     for j from 1 to n do
         m[i,j]:=0;
  od;od;
  for i from 1 to n do
        m[i,i]:=3;
  od;
  for i from 2 to n do
        m[i,i-1]:=-1;
  od;
  for i from 1 to n-1 do
        m[i,i+1]:=-1;
  od;
  m[1,n]:=-1;
```

```
  m[n,1]:=-1;
  m;
  end;
```

roue := proc(n)
local m, i, j;
    m := array(1 .. n, 1 .. n);
    for i to n do for j to n do m[i, j] := 0 end do end do;
    for i to n do m[i, i] := 3 end do;
    for i from 2 to n do m[i, i − 1] := -1 end do;
    for i to n − 1 do m[i, i + 1] := -1 end do;
    m[1, n] := -1;
    m[n, 1] := -1;
    m
end proc

```
> evalm(roue(5));
```

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

```
> m:=evalm(roue(6));
```

$$m := \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & 0 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & -1 & 3 \end{bmatrix}$$

```
> recurrentes(m,[3,3,3,3,3,3]);
```

[[0, 1, 1, 1, 1, 2], [0, 1, 1, 1, 2, 1], [0, 1, 1, 1, 2, 2], [0, 1, 1, 2, 0, 2], [0, 1, 1, 2, 1, 1],
[0, 1, 1, 2, 1, 2], [0, 1, 1, 2, 2, 1], [0, 1, 1, 2, 2, 2], [0, 1, 2, 0, 1, 2], [0, 1, 2, 0, 2, 1],
[0, 1, 2, 0, 2, 2], [0, 1, 2, 1, 0, 2], [0, 1, 2, 1, 1, 1], [0, 1, 2, 1, 1, 2], [0, 1, 2, 1, 2, 1],
[0, 1, 2, 1, 2, 2], [0, 1, 2, 2, 0, 2], [0, 1, 2, 2, 1, 1], [0, 1, 2, 2, 1, 2], [0, 1, 2, 2, 2, 1],
[0, 1, 2, 2, 2, 2], [0, 2, 0, 1, 1, 2], [0, 2, 0, 1, 2, 1], [0, 2, 0, 1, 2, 2], [0, 2, 0, 2, 0, 2],
[0, 2, 0, 2, 1, 1], [0, 2, 0, 2, 1, 2], [0, 2, 0, 2, 2, 1], [0, 2, 0, 2, 2, 2], [0, 2, 1, 0, 1, 2],
[0, 2, 1, 0, 2, 1], [0, 2, 1, 0, 2, 2], [0, 2, 1, 1, 0, 2], [0, 2, 1, 1, 1, 1], [0, 2, 1, 1, 1, 2],
[0, 2, 1, 1, 2, 1], [0, 2, 1, 1, 2, 2], [0, 2, 1, 2, 0, 2], [0, 2, 1, 2, 1, 1], [0, 2, 1, 2, 1, 2],
[0, 2, 1, 2, 2, 1], [0, 2, 1, 2, 2, 2], [0, 2, 2, 0, 1, 2], [0, 2, 2, 0, 2, 1], [0, 2, 2, 0, 2, 2],
[0, 2, 2, 1, 0, 2], [0, 2, 2, 1, 1, 1], [0, 2, 2, 1, 1, 2], [0, 2, 2, 1, 2, 1], [0, 2, 2, 1, 2, 2],

[0, 2, 2, 2, 0, 2], [0, 2, 2, 2, 1, 1], [0, 2, 2, 2, 1, 2], [0, 2, 2, 2, 2, 1], [0, 2, 2, 2, 2, 2],
[1, 0, 1, 1, 1, 2], [1, 0, 1, 1, 2, 1], [1, 0, 1, 1, 2, 2], [1, 0, 1, 2, 0, 2], [1, 0, 1, 2, 1, 1],
[1, 0, 1, 2, 1, 2], [1, 0, 1, 2, 2, 1], [1, 0, 1, 2, 2, 2], [1, 0, 2, 0, 1, 2], [1, 0, 2, 0, 2, 1],
[1, 0, 2, 0, 2, 2], [1, 0, 2, 1, 0, 2], [1, 0, 2, 1, 1, 1], [1, 0, 2, 1, 1, 2], [1, 0, 2, 1, 2, 1],
[1, 0, 2, 1, 2, 2], [1, 0, 2, 2, 0, 2], [1, 0, 2, 2, 1, 1], [1, 0, 2, 2, 1, 2], [1, 0, 2, 2, 2, 1],
[1, 0, 2, 2, 2, 2], [1, 1, 0, 1, 1, 2], [1, 1, 0, 1, 2, 1], [1, 1, 0, 1, 2, 2], [1, 1, 0, 2, 0, 2],
[1, 1, 0, 2, 1, 1], [1, 1, 0, 2, 1, 2], [1, 1, 0, 2, 2, 1], [1, 1, 0, 2, 2, 2], [1, 1, 1, 0, 1, 2],
[1, 1, 1, 0, 2, 1], [1, 1, 1, 0, 2, 2], [1, 1, 1, 1, 0, 2], [1, 1, 1, 1, 1, 2], [1, 1, 1, 1, 2, 0],
[1, 1, 1, 1, 2, 1], [1, 1, 1, 1, 2, 2], [1, 1, 1, 2, 0, 1], [1, 1, 1, 2, 0, 2], [1, 1, 1, 2, 1, 0],
[1, 1, 1, 2, 1, 1], [1, 1, 1, 2, 1, 2], [1, 1, 1, 2, 2, 0], [1, 1, 1, 2, 2, 1], [1, 1, 1, 2, 2, 2],
[1, 1, 2, 0, 1, 1], [1, 1, 2, 0, 1, 2], [1, 1, 2, 0, 2, 0], [1, 1, 2, 0, 2, 1], [1, 1, 2, 0, 2, 2],
[1, 1, 2, 1, 0, 1], [1, 1, 2, 1, 0, 2], [1, 1, 2, 1, 1, 0], [1, 1, 2, 1, 1, 1], [1, 1, 2, 1, 1, 2],
[1, 1, 2, 1, 2, 0], [1, 1, 2, 1, 2, 1], [1, 1, 2, 1, 2, 2], [1, 1, 2, 2, 0, 1], [1, 1, 2, 2, 0, 2],
[1, 1, 2, 2, 1, 0], [1, 1, 2, 2, 1, 1], [1, 1, 2, 2, 1, 2], [1, 1, 2, 2, 2, 0], [1, 1, 2, 2, 2, 1],
[1, 1, 2, 2, 2, 2], [1, 2, 0, 1, 1, 1], [1, 2, 0, 1, 1, 2], [1, 2, 0, 1, 2, 0], [1, 2, 0, 1, 2, 1],
[1, 2, 0, 1, 2, 2], [1, 2, 0, 2, 0, 1], [1, 2, 0, 2, 0, 2], [1, 2, 0, 2, 1, 0], [1, 2, 0, 2, 1, 1],
[1, 2, 0, 2, 1, 2], [1, 2, 0, 2, 2, 0], [1, 2, 0, 2, 2, 1], [1, 2, 0, 2, 2, 2], [1, 2, 1, 0, 1, 1],
[1, 2, 1, 0, 1, 2], [1, 2, 1, 0, 2, 0], [1, 2, 1, 0, 2, 1], [1, 2, 1, 0, 2, 2], [1, 2, 1, 1, 0, 1],
[1, 2, 1, 1, 0, 2], [1, 2, 1, 1, 1, 0], [1, 2, 1, 1, 1, 1], [1, 2, 1, 1, 1, 2], [1, 2, 1, 1, 2, 0],
[1, 2, 1, 1, 2, 1], [1, 2, 1, 1, 2, 2], [1, 2, 1, 2, 0, 1], [1, 2, 1, 2, 0, 2], [1, 2, 1, 2, 1, 0],
[1, 2, 1, 2, 1, 1], [1, 2, 1, 2, 1, 2], [1, 2, 1, 2, 2, 0], [1, 2, 1, 2, 2, 1], [1, 2, 1, 2, 2, 2],
[1, 2, 2, 0, 1, 1], [1, 2, 2, 0, 1, 2], [1, 2, 2, 0, 2, 0], [1, 2, 2, 0, 2, 1], [1, 2, 2, 0, 2, 2],
[1, 2, 2, 1, 0, 1], [1, 2, 2, 1, 0, 2], [1, 2, 2, 1, 1, 0], [1, 2, 2, 1, 1, 1], [1, 2, 2, 1, 1, 2],
[1, 2, 2, 1, 2, 0], [1, 2, 2, 1, 2, 1], [1, 2, 2, 1, 2, 2], [1, 2, 2, 2, 0, 1], [1, 2, 2, 2, 0, 2],
[1, 2, 2, 2, 1, 0], [1, 2, 2, 2, 1, 1], [1, 2, 2, 2, 1, 2], [1, 2, 2, 2, 2, 0], [1, 2, 2, 2, 2, 1],
[1, 2, 2, 2, 2, 2], [2, 0, 1, 1, 1, 1], [2, 0, 1, 1, 1, 2], [2, 0, 1, 1, 2, 0], [2, 0, 1, 1, 2, 1],
[2, 0, 1, 1, 2, 2], [2, 0, 1, 2, 0, 1], [2, 0, 1, 2, 0, 2], [2, 0, 1, 2, 1, 0], [2, 0, 1, 2, 1, 1],
[2, 0, 1, 2, 1, 2], [2, 0, 1, 2, 2, 0], [2, 0, 1, 2, 2, 1], [2, 0, 1, 2, 2, 2], [2, 0, 2, 0, 1, 1],
[2, 0, 2, 0, 1, 2], [2, 0, 2, 0, 2, 0], [2, 0, 2, 0, 2, 1], [2, 0, 2, 0, 2, 2], [2, 0, 2, 1, 0, 1],
[2, 0, 2, 1, 0, 2], [2, 0, 2, 1, 1, 0], [2, 0, 2, 1, 1, 1], [2, 0, 2, 1, 1, 2], [2, 0, 2, 1, 2, 0],
[2, 0, 2, 1, 2, 1], [2, 0, 2, 1, 2, 2], [2, 0, 2, 2, 0, 1], [2, 0, 2, 2, 0, 2], [2, 0, 2, 2, 1, 0],
[2, 0, 2, 2, 1, 1], [2, 0, 2, 2, 1, 2], [2, 0, 2, 2, 2, 0], [2, 0, 2, 2, 2, 1], [2, 0, 2, 2, 2, 2],
[2, 1, 0, 1, 1, 1], [2, 1, 0, 1, 1, 2], [2, 1, 0, 1, 2, 0], [2, 1, 0, 1, 2, 1], [2, 1, 0, 1, 2, 2],
[2, 1, 0, 2, 0, 1], [2, 1, 0, 2, 0, 2], [2, 1, 0, 2, 1, 0], [2, 1, 0, 2, 1, 1], [2, 1, 0, 2, 1, 2],
[2, 1, 0, 2, 2, 0], [2, 1, 0, 2, 2, 1], [2, 1, 0, 2, 2, 2], [2, 1, 1, 0, 1, 1], [2, 1, 1, 0, 1, 2],
[2, 1, 1, 0, 2, 0], [2, 1, 1, 0, 2, 1], [2, 1, 1, 0, 2, 2], [2, 1, 1, 1, 0, 1], [2, 1, 1, 1, 0, 2],
[2, 1, 1, 1, 1, 0], [2, 1, 1, 1, 1, 1], [2, 1, 1, 1, 1, 2], [2, 1, 1, 1, 2, 0], [2, 1, 1, 1, 2, 1],
[2, 1, 1, 1, 2, 2], [2, 1, 1, 2, 0, 1], [2, 1, 1, 2, 0, 2], [2, 1, 1, 2, 1, 0], [2, 1, 1, 2, 1, 1],
[2, 1, 1, 2, 1, 2], [2, 1, 1, 2, 2, 0], [2, 1, 1, 2, 2, 1], [2, 1, 1, 2, 2, 2], [2, 1, 2, 0, 1, 1],
[2, 1, 2, 0, 1, 2], [2, 1, 2, 0, 2, 0], [2, 1, 2, 0, 2, 1], [2, 1, 2, 0, 2, 2], [2, 1, 2, 1, 0, 1],
[2, 1, 2, 1, 0, 2], [2, 1, 2, 1, 1, 0], [2, 1, 2, 1, 1, 1], [2, 1, 2, 1, 1, 2], [2, 1, 2, 1, 2, 0],
[2, 1, 2, 1, 2, 1], [2, 1, 2, 1, 2, 2], [2, 1, 2, 2, 0, 1], [2, 1, 2, 2, 0, 2], [2, 1, 2, 2, 1, 0],
[2, 1, 2, 2, 1, 1], [2, 1, 2, 2, 1, 2], [2, 1, 2, 2, 2, 0], [2, 1, 2, 2, 2, 1], [2, 1, 2, 2, 2, 2],

[2, 2, 0, 1, 1, 1], [2, 2, 0, 1, 1, 2], [2, 2, 0, 1, 2, 0], [2, 2, 0, 1, 2, 1], [2, 2, 0, 1, 2, 2],
[2, 2, 0, 2, 0, 1], [2, 2, 0, 2, 0, 2], [2, 2, 0, 2, 1, 0], [2, 2, 0, 2, 1, 1], [2, 2, 0, 2, 1, 2],
[2, 2, 0, 2, 2, 0], [2, 2, 0, 2, 2, 1], [2, 2, 0, 2, 2, 2], [2, 2, 1, 0, 1, 1], [2, 2, 1, 0, 1, 2],
[2, 2, 1, 0, 2, 0], [2, 2, 1, 0, 2, 1], [2, 2, 1, 0, 2, 2], [2, 2, 1, 1, 0, 1], [2, 2, 1, 1, 0, 2],
[2, 2, 1, 1, 1, 0], [2, 2, 1, 1, 1, 1], [2, 2, 1, 1, 1, 2], [2, 2, 1, 1, 2, 0], [2, 2, 1, 1, 2, 1],
[2, 2, 1, 1, 2, 2], [2, 2, 1, 2, 0, 1], [2, 2, 1, 2, 0, 2], [2, 2, 1, 2, 1, 0], [2, 2, 1, 2, 1, 1],
[2, 2, 1, 2, 1, 2], [2, 2, 1, 2, 2, 0], [2, 2, 1, 2, 2, 1], [2, 2, 1, 2, 2, 2], [2, 2, 2, 0, 1, 1],
[2, 2, 2, 0, 1, 2], [2, 2, 2, 0, 2, 0], [2, 2, 2, 0, 2, 1], [2, 2, 2, 0, 2, 2], [2, 2, 2, 1, 0, 1],
[2, 2, 2, 1, 0, 2], [2, 2, 2, 1, 1, 0], [2, 2, 2, 1, 1, 1], [2, 2, 2, 1, 1, 2], [2, 2, 2, 1, 2, 0],
[2, 2, 2, 1, 2, 1], [2, 2, 2, 1, 2, 2], [2, 2, 2, 2, 0, 1], [2, 2, 2, 2, 0, 2], [2, 2, 2, 2, 1, 0],
[2, 2, 2, 2, 1, 1], [2, 2, 2, 2, 1, 2], [2, 2, 2, 2, 2, 0], [2, 2, 2, 2, 2, 1], [2, 2, 2, 2, 2, 2]]

```
> nops(%);
```

$$320$$

```
> det(m);
```

$$320$$

```
> u:='u';v:='v';s:=linalg[ismith](m,u,v);
```

$$u := u$$

$$v := v$$

$$s := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 40 \end{bmatrix}$$

```
>
> evalm(u^(-1));
```

$$\begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -3 & 13 & 2 \\ 0 & -1 & 3 & 8 & -7 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> epsilon(m);
```

$$[4, 4, 4, 4, 4, 4]$$

```
> avalanche(%,m);
```

$$[2, 2, 2, 2, 2, 2]$$

```
> u1:=avalanche([10,23,3,10,10,10],m);
```

$$u1 := [2, 2, 1, 0, 1, 2]$$

```
> u2:=avalanche([4,6,3,4,4,4],m);
```

$$u2 := [2, 0, 1, 1, 1, 1]$$

```
> v:=u1;for i from 2 to 8 do v:=avalanche(u1+v,m) od;
```

$$v := [2, 2, 1, 0, 1, 2]$$
$$v := [2, 1, 1, 2, 1, 1]$$
$$v := [2, 0, 2, 1, 2, 0]$$
$$v := [0, 2, 2, 0, 2, 2]$$
$$v := [0, 1, 2, 2, 2, 1]$$
$$v := [1, 2, 2, 1, 2, 2]$$
$$v := [1, 2, 0, 2, 0, 2]$$
$$v := [2, 2, 2, 2, 2, 2]$$

```
> v:=u2;for i from 2 to 40 do v:=avalanche(u2+v,m) od;
```

$$v := [2, 0, 1, 1, 1, 1]$$
$$v := [2, 2, 0, 1, 1, 1]$$
$$v := [0, 2, 0, 1, 1, 2]$$
$$v := [1, 1, 0, 1, 1, 2]$$
$$v := [1, 2, 2, 0, 1, 2]$$
$$v := [2, 1, 2, 0, 1, 2]$$
$$v := [1, 1, 2, 0, 2, 0]$$
$$v := [2, 0, 2, 0, 2, 0]$$
$$v := [2, 2, 1, 0, 2, 0]$$
$$v := [0, 2, 1, 0, 2, 1]$$
$$v := [1, 1, 1, 0, 2, 1]$$
$$v := [1, 2, 2, 2, 1, 1]$$
$$v := [2, 1, 2, 2, 1, 1]$$
$$v := [0, 1, 2, 2, 1, 2]$$
$$v := [1, 0, 2, 2, 1, 2]$$
$$v := [1, 2, 1, 2, 1, 2]$$
$$v := [2, 1, 1, 2, 1, 2]$$
$$v := [1, 1, 1, 2, 2, 0]$$
$$v := [2, 0, 1, 2, 2, 0]$$
$$v := [2, 2, 0, 2, 2, 0]$$
$$v := [0, 2, 0, 2, 2, 1]$$
$$v := [1, 1, 0, 2, 2, 1]$$
$$v := [1, 2, 2, 1, 2, 1]$$
$$v := [2, 1, 2, 1, 2, 1]$$
$$v := [0, 1, 2, 1, 2, 2]$$
$$v := [1, 0, 2, 1, 2, 2]$$
$$v := [1, 2, 1, 1, 2, 2]$$
$$v := [2, 1, 1, 1, 2, 2]$$
$$v := [1, 1, 1, 2, 0, 1]$$
$$v := [2, 0, 1, 2, 0, 1]$$

$$v := [2, 2, 0, 2, 0, 1]$$
$$v := [0, 2, 0, 2, 0, 2]$$
$$v := [1, 1, 0, 2, 0, 2]$$
$$v := [1, 2, 2, 1, 0, 2]$$
$$v := [2, 1, 2, 1, 0, 2]$$
$$v := [1, 1, 2, 1, 1, 0]$$
$$v := [2, 0, 2, 1, 1, 0]$$
$$v := [2, 2, 1, 1, 1, 0]$$
$$v := [0, 2, 1, 1, 1, 1]$$
$$v := [2, 2, 2, 2, 2, 2]$$

>