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

```
2  non on n'a pas utilis{'e} le fait que b soit reduite pour
d(L)<=prod_{i=1}^n ||b_i||, c'est l'in{'e}alit{'e} d'Hadamard
```

```
3  C'est deja programme dans maple: l'instruction lattice existe deja
dans maple7 et superieur. Dans maple9 il y a en plus le paquet with(IntegerRelations);
qui contient l'instruction LLC. Pour xcas il y a l'instruction: lll, et
l·instruction pari_qflll. (qui retourne la matrice de passage.)
```

```
4  M:=matrix([[1,2,3],[-1,0,1],[0,1,1]]);
```

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

```
5  ortho:=proc(A)//On teste d'abord l'orthogonalisation des lignes.
local b,bb,mu;
b:=[seq((A)[i],i=1..rowdim(A))];
n:=nops(b);//la dimension du reseau
bb:=b;
for i from 2 to n do
bb[i]:=normal(b[i]-add((b[i]*bb[j])/(bb[j]*bb[j])*bb[j],j=1..i-1));
od;
mu:=matrix(n,n,(i,j)->(b[i]*bb[j])/(bb[j]*bb[j]));
mu,bb;
end:proc:
// Warning: b bb  declared as global variable(s)
// Warning: i n j  declared as global variable(s)
// End defining ortho
```

```
proc(A)
  local b,bb,mu;
  b:=[seq(A[i],i=(1 .. (rowdim(A))))];
  n:=nops(b);
  bb:=b;
  for i from 2 to n+1/2 do
  bb[i]:=normal(b[i]-add((b[i]*bb[j])/(bb[j]*bb[j])*bb[j],j=(1 .. (i-1))));
   od;;
  mu:=matrix(n,n, (i,j)->(b[i]*bb[j])/(bb[j]*bb[j]));
  mu,bb;

end;
```

```
6  v:=ortho(M)[2];v[1]*v[2],v[2]*v[3],v[3]*v[1];
```

$$\left(\begin{bmatrix} 1 & 2 & 3 \\ \frac{-8}{7} & \frac{-2}{7} & \frac{4}{7} \\ \frac{-1}{6} & \frac{1}{3} & \frac{-1}{6} \end{bmatrix}, \ 0, \ 0, \ 0 \right)$$

```
7
```

8 Prog Edit Add | nxt | OK | Save

```
myLLL:= proc(A)//notre LLL pour les lignes;
 local b,mu,bb,BB,k,r;
b:=A;//les vecteurs sont les lignes de A
n:=nops(b); //la dimension du reseau
bb:=b;
for i from 2 to n do
bb[i]:=normal(b[i]-add((b[i]*bb[j])/(bb[j]*bb[j])*bb[j],j=1..i-1));
od;
mu:=matrix(n,n,(i,j)->(b[i]*bb[j])/(bb[j]*bb[j]));
BB:=[seq(bb[i]*bb[i],i=1..n)];  //BB= les carre des normes des b^*
k:=2;
while ( k<n+1)
do
if abs(mu[k,k-1])> 0.5  then
       r:=round(mu[k,k-1]);
       b[k]:=b[k]-r*b[k-1];
        for j from 1 to k-2 do mu[k,j]:=mu[k,j]-r*mu[k-1,j];  od;
       mu[k,k-1]:=mu[k,k-1]-r;
fi;
if ( BB[k] < BB[k-1]*(3/4-mu[k,k-1]^2))
then
mk:=mu[k,k-1]; //on sauve l'ancienne valeur de mu[k,k-1]
 cc:=BB[k]+mk^2*BB[k-1];    //cc=norme de c^*_{k-1} au carre
mu[k,k-1]:=mk*BB[k-1]/cc;  //on remplace mu[k,k-1] par la valeur de nu_{k,k-1}
for i from k+1 to n do
mik:=mu[i,k-1]; //on sauvegarde
mu[i,k-1]:=mu[i,k-1]*mu[k,k-1]+mu[i,k]*BB[k]/cc;
mu[i,k]:=mik-mu[i,k]*mk;
od;
for j from 1 to k-2 do
mkj:=mu[k,j];mu[k,j]:=mu[k-1,j];mu[k-1,j]:=mkj;
od;
//on echange b_k et b_{k-1}
c:=b[k];b[k]:=b[k-1];b[k-1]:=c;
BB[k]:=BB[k-1]*BB[k]/cc;BB[k-1]:=cc;   //on corrige le carre des normes de b^*
k:=max(2,k-1);
else
l:=k-2;
while l>0 do
   if abs(mu[k,l])>0.5  then
   r:=round(mu[k,l]);b[k]:=b[k]-r*b[l];
    for j from 1 to l-1 do mu[k,j]:=mu[k,j]-r*mu[l,j]   od;
   mu[k,l]:=mu[k,l]-r;
    fi;
   l:=l-1;
    od;
k:=k+1;
fi;
od;
b;
end proc:
```

```
// Warning: b bb  declared as global variable(s)
// Warning: n i j mk cc mik mkj c l  declared as global variable(s)
// End defining myLLL
```

Done Menu

9 sac {'a} dos:
On cree la clef publique: A (suite supercroissante)

10
```
s:=rand(200):A:=[s]:for i from 1 to 6 do s:=s+rand(200):A:=[op(A),s]: od;
```

(Done, Done, [137 168 172 189 301 344 397]) Menu

11
```
add(A);#on prend superieur a la somme
```

1708 Menu

12
```
M:=4001;w:=1001;igcd(w,M);
```

(4001, 1001, 1) Menu

13
```
B:=[seq(w*i mod M,i=A)];
```

[1103 126 129 1142 1226 258 1298] Menu

14
```
E:=[1,0,1,0,0,0,1];// le message a transmettre.
```

[1 0 1 0 0 0 1] Menu

15
```
c:=E*B;//le message crypt{'e}
```

16 M:=matrix(nops(B)+1,nops(B)+1,(i,j)->if i=j-1 then 1 else 0 fi);

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

Menu

17 M[nops(B)+1,1]:=c;for i from 1 to nops(B) do M[nops(B)+1,i+1]:=B[i] od:M;

$$\left(\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2530 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \text{Done}, \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2530 & 1103 & 126 & 129 & 1142 & 1226 & 258 & 1298 \end{bmatrix} \right)$$

Menu

18 myLLL(transpose(M));// la premiere ligne est bien le message E

Evaluation time: 0.4375

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

Menu

19 //un autre essai:

Syntax compatibility mode maple
Parse error line 2 at/

undef

Menu

20 s:=rand(200):A:=[s]:for i from 1 to 20 do s:=s+rand(200):A:=[op(A),s]: od:

(Done, Done, Done)

Menu

21 add(A);

18790

Menu

22 M:=40001;w:=2357;igcd(w,M);

(40001, 2357, 1)

Menu

23 B:=[seq(w*i mod M,i=A)];

Menu

24 E:=[seq(rand(2)*rand(2),i=0..nops(A))]:// le message a transmettre. On met un carre pour avoir pas mal de zeros

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

Menu

25 c:=E*B://le message crypt{'e}

137345

Menu

26 M:=matrix(nops(B)+1,nops(B)+1,(i,j)->if i=j-1 then 1 else 0 fi):

// Success

Done

```
27 M[nops(B)+1,1]:=c:for i from 1 to nops(B) do M[nops(B)+1,i+1]:=B[i] od:
                                   ( Done, Done )
28 myLLL(transpose(M));
Evaluation time: 17.0859
```

-1	0	-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	-1	0	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0
-1	-1	0	-1	-1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	-1	0	0	-1	1	-1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	1	0	-1	-1	0	0	0	0	0	0	0	0	0
0	0	1	-1	0	0	-1	0	1	1	-1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	-1	1	-1	0	-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	0	0	-1	0	1	-1	1	0	-1	1	0	0	0	0	0	0	0	0	0
-1	1	0	0	0	0	-1	0	0	0	-1	0	0	0	0	1	0	0	0	0	0	0
0	-1	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	1	0	-1	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	-1	0	0	1	-1	1	0	-1	0	0	0	0	0	0
0	0	0	0	-1	0	0	0	0	-1	0	-1	1	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	-1	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	-1	0	0	0	0	1	-1	0	-1	0	0	0	0	1	0	0	0	0
0	0	-1	1	0	0	-1	0	0	0	1	0	0	1	-1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	-1	0	-1	1	0	0	0	-1	0	0	0	1	0	0	0
0	0	0	0	-1	0	1	0	-1	1	0	-1	-1	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	1	0	-1	0	-1	0	0	0	-1	0	0	0	0	1	0
0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	1	1	0	0	1	0
1	-2	0	-1	2	-4	-3	0	-2	-2	1	1	-3	-1	3	1	4	0	3	1	0	-5

```
29 lll(transpose(M));
Evaluation time: 2.16406
```

-1	0	-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	-1	0	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
-1	-1	0	-1	-1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	-1	0	0	-1	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	1	0	-1	-1	0	0	0	0	0	0	0	0	0	0
0	0	1	-1	0	0	-1	0	1	1	-1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	-1	0	0	0	-1	0	1	1	0	-1	0	0	0	0	0	0	0	0	0	1
-1	0	0	0	0	-1	0	1	-1	1	0	-1	1	0	0	0	0	0	0	0	0	0	-1

```
-1  1  0  0  0  0 -1  0  0  0 -1  0  0  0  0  1  0  0  0  0  0  0 |  0 -1
 0 -1  0  0  0  0  0  0  0  1 -1 -1  0  0  0  1  0  0  0  0  0  0 |  0  0
 1  0  0  0  1  0  0  0  0  0  1  0 -1  0  0  1  0  0  0  0  0  1 |  0  1
 0  0  1  0  1  0  0 -1  0  0  1 -1  1  0 -1  0  0  0  0  0  0  0 |  0  0
 0  0  0  0 -1  0  0  0  0 -1  0 -1  1  0  1  0  0  0  0  0  0  0 |  0  0
 0  0  0  1  0  0  0  1 -1  0  0  0  0  0  0  0  1  0  0  0  0  1 |  0  0
```

Menu

30 pari();

```
All PARI functions are now defined with the pari_ prefix.
PARI functions are also defined without prefix except:
abs acos acosh arg asin asinh atan atanh binomial bitand bitor bitxor ceil charpoly concat conj content cos cosh divis
Note that p-adic numbers must have O argument quoted e.g. 905/7+O('7^3')
Type ?pari for short help
Inside xcas, try Help->Manuals->PARI for HTML help
```

Menu

31 pari le fait sur les colonnes, et retourne la matrice de passage entre les 2 bases

32 transpose(M*pari_qflll(M));

```
-1  0 -1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0 -1  0 -1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
 0  0  0  0  1 -1  0  0  0  0 -1  0  1  0  0  0  0  0  0  0  0
 0  0  0  1  0  0  0  1  0  1  0 -1 -1  0  0  0  0  0  0  0  0
 0  0  0  1 -1  0  0 -1  1 -1  1  0  0  0  0  0  0  0  0  0  0
 0  0  1 -1  0  0  1  0  0  0 -1  0  0 -1  1  0  0  0  0  0  0
-1  0  0  1  0 -1  0  0  0  1  1  0  0  0 -1  0  0  0  0  0  0
 0  0  0  0 -1  0  0  0  0 -1  0 -1  1  0  1  0  0  0  0  0  0
 0 -1  0  0  1  0  0  0  1 -1  0  0  0  1 -1  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0 -1  1  1 -1  0  0  0  0
 0  0  0  0 -1  0  0  0  1 -1 -1  0  1 -1  0  0  1  0  0  0  0
-1  0  0  0  0  0  0  0  0  1  0  1 -1  0  1  0  0  0  0  0  1
 1  0 -1  0  0 -1  0  0  1  0  0 -1  0  1  0  0  0  0  0  0  0
 1  0  0  0  0  0  0  0  0 -1  1 -1  0  1  0  0  1  0  0  0  1
 0  0  1 -1  0  0  0 -1  0  0 -1  0  0  0  0  0  0  1  0  0  0
 0  1 -1  1  0 -1  0  0  0  0  0  0  0  0 -1  0  0  0  1  0  0
-1  0  0  0  0  0  0  0  0  0  0  0  1 -1  0 -1  0  0  1  0  0
 0  1  0  1  1 -1  0  0  0  0  0  0  0 -1 -1  0  0  0  0  1  0
 1  0  0  0  0  1  0 -1  0  0  0  0  0  0  0  0  1 -1  0  0  0
 0  0  0  0  0  0  0  0 -1  1 -1  0  0 -1  0  0  0  0  0  0  1
 0  0  0  0  0  0  0 -1  0 -1 -1  0  0 -1  0 -1  0  0  0 -1 -1
 0 -3  1  0  0 -4 -3  0 -1 -2  0  0 -3 -1  3  1  4  0  3  2  0 -5
```

Menu

33 On essaye un autre reseau qui contient aussi une solution. Attention, le sac a dos modulaire peut avoir plusieurs so
c'est le super croissant qui n'en a qu'une

```
34 r:=-floor(-sqrt(nops(B))/2);M[nops(B)+1]:=r*M[nops(B)+1]:
   for i from 1 to nops(B) do M[i,1]:=1/2 od:M;
```

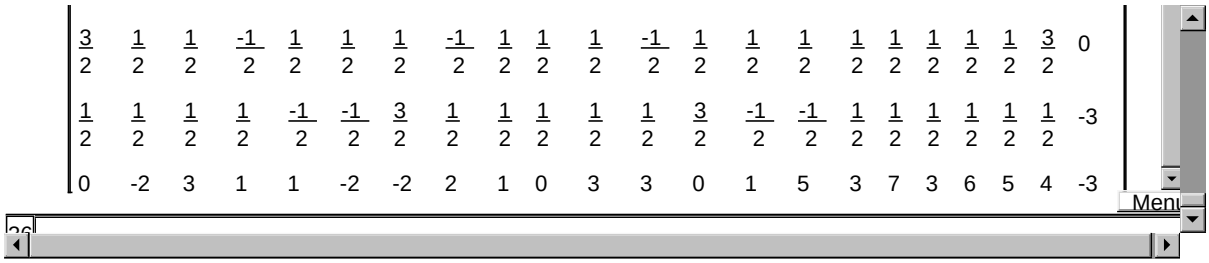
$$\begin{bmatrix}
\frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2}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

3, Done, Done,

Menu

```
35 myLLL(transpose(M))+matrix(nops(B)+1,nops(B)+1,(i,j)->if j=nops(B)+1 then 0 else 1/2 fi);
   // Warning: B  declared as global variable(s)
   Evaluation time: 21.9531
```

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


3/2 1/2 1/2 -1/2 1/2 1/2 1/2 -1/2 1/2 1/2 1/2 -1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 3/2 0
1/2 1/2 1/2 1/2 -1/2 -1/2 3/2 1/2 1/2 1/2 1/2 1/2 3/2 -1/2 -1/2 1/2 1/2 1/2 1/2 1/2 1/2 -3
0 -2 3 1 1 -2 -2 2 1 0 3 3 0 1 5 3 7 3 6 5 4 -3
Menu