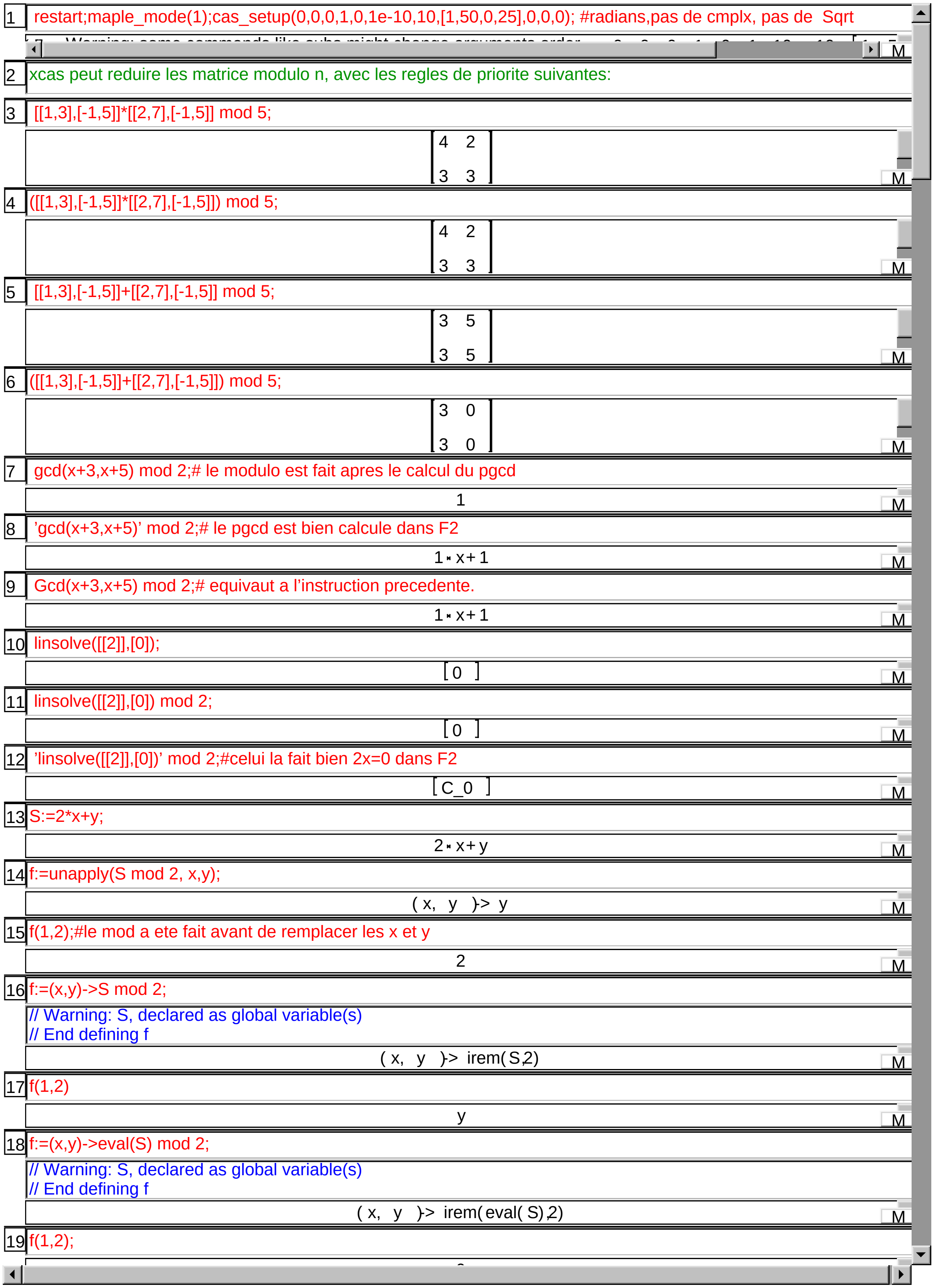
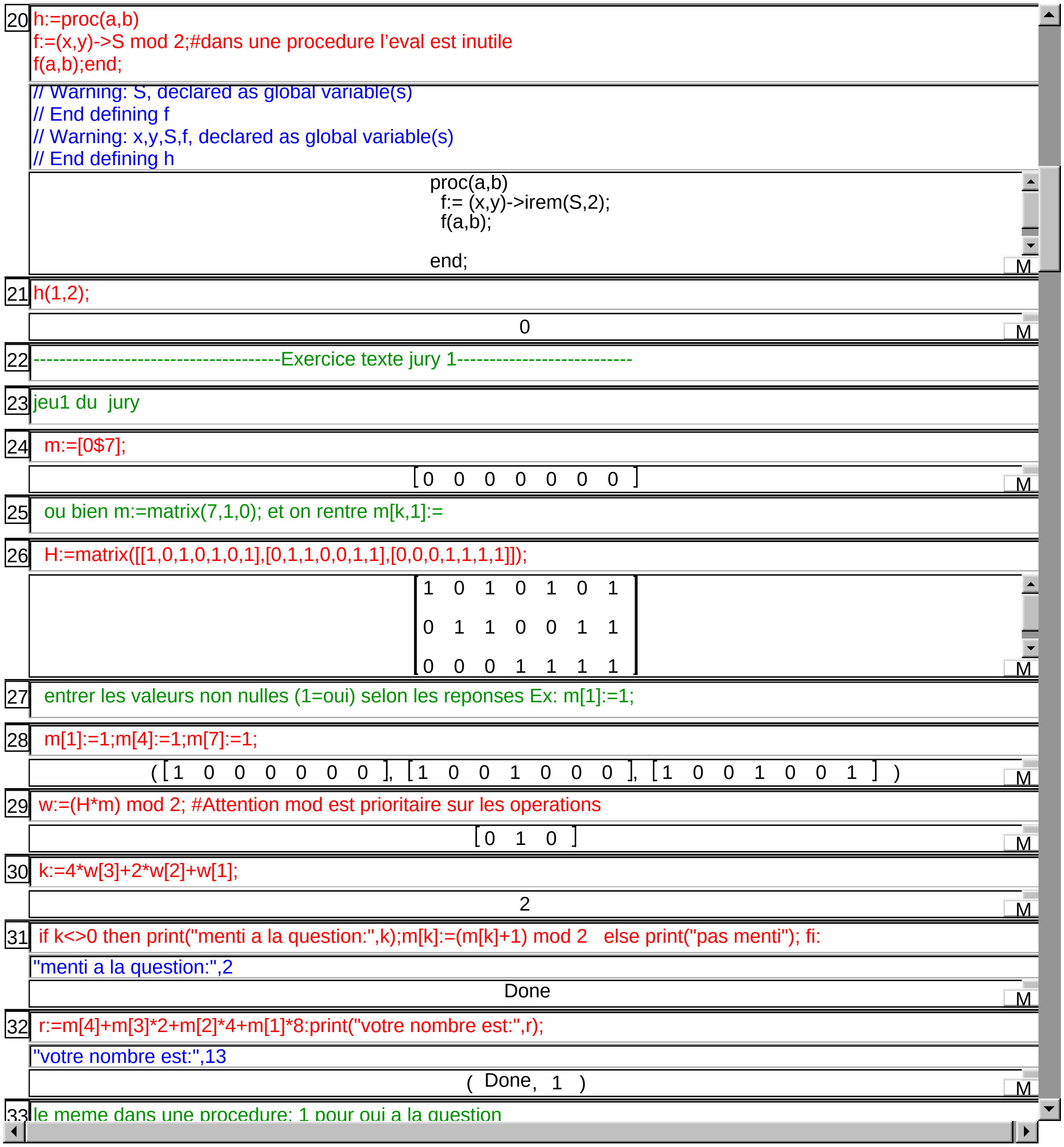


```
1  restart;maple_mode(1);cas_setup(0,0,0,1,0,1e-10,10,[1,50,0,25],0,0,0); #radians,pas de cmplx, pas de  Sqrt
   Warning: some commands like subs might change arguments order
2  xcas peut reduire les matrice modulo n, avec les regles de priorite suivantes:
3   [[1,3],[-1,5]]*[[2,7],[-1,5]] mod 5;
                                   [ 4  2 ]
                                   [ 3  3 ]
4  ([[1,3],[-1,5]]*[[2,7],[-1,5]]) mod 5;
                                   [ 4  2 ]
                                   [ 3  3 ]
5   [[1,3],[-1,5]]+[[2,7],[-1,5]] mod 5;
                                   [ 3  5 ]
                                   [ 3  5 ]
6  ([[1,3],[-1,5]]+[[2,7],[-1,5]]) mod 5;
                                   [ 3  0 ]
                                   [ 3  0 ]
7   gcd(x+3,x+5) mod 2;# le modulo est fait apres le calcul du pgcd
                                      1
8   'gcd(x+3,x+5)' mod 2;# le pgcd est bien calcule dans F2
                                   1·x+1
9   Gcd(x+3,x+5) mod 2;# equivaut a l'instruction precedente.
                                   1·x+1
10  linsolve([[2]],[0]);
                                    [0]
11  linsolve([[2]],[0]) mod 2;
                                    [0]
12  'linsolve([[2]],[0])' mod 2;#celui la fait bien 2x=0 dans F2
                                   [C_0]
13 S:=2*x+y;
                                   2·x+y
14 f:=unapply(S mod 2, x,y);
                                (x, y)-> y
15 f(1,2);#le mod a ete fait avant de remplacer les x et y
                                      2
16 f:=(x,y)->S mod 2;
   // Warning: S, declared as global variable(s)
   // End defining f
                             (x, y)-> irem(S,2)
17 f(1,2)
                                      y
18 f:=(x,y)->eval(S) mod 2;
   // Warning: S, declared as global variable(s)
   // End defining f
                          (x, y)-> irem(eval(S),2)
19 f(1,2);
```



20
```
h:=proc(a,b)
f:=(x,y)->S mod 2;#dans une procedure l'eval est inutile
f(a,b);end;
```

```
// Warning: S, declared as global variable(s)
// End defining f
// Warning: x,y,S,f, declared as global variable(s)
// End defining h
```

```
proc(a,b)
  f:= (x,y)->irem(S,2);
  f(a,b);

end;
```

21
```
h(1,2);
```

```
0
```

22 ------------------------------------Exercice texte jury 1--------------------------

23 jeu1 du jury

24
```
m:=[0$7];
```

```
[0  0  0  0  0  0  0 ]
```

25 ou bien m:=matrix(7,1,0); et on rentre m[k,1]:=

26
```
H:=matrix([[1,0,1,0,1,0,1],[0,1,1,0,0,1,1],[0,0,0,1,1,1,1]]);
```

```
1  0  1  0  1  0  1

0  1  1  0  0  1  1

0  0  0  1  1  1  1
```

27 entrer les valeurs non nulles (1=oui) selon les reponses Ex: m[1]:=1;

28
```
m[1]:=1;m[4]:=1;m[7]:=1;
```

```
([1  0  0  0  0  0  0 ], [1  0  0  1  0  0  0 ], [1  0  0  1  0  0  1 ] )
```

29
```
w:=(H*m) mod 2; #Attention mod est prioritaire sur les operations
```

```
[0  1  0 ]
```

30
```
k:=4*w[3]+2*w[2]+w[1];
```

```
2
```

31
```
if k<>0 then print("menti a la question:",k);m[k]:=(m[k]+1) mod 2   else print("pas menti"); fi:
```

```
"menti a la question:",2
```

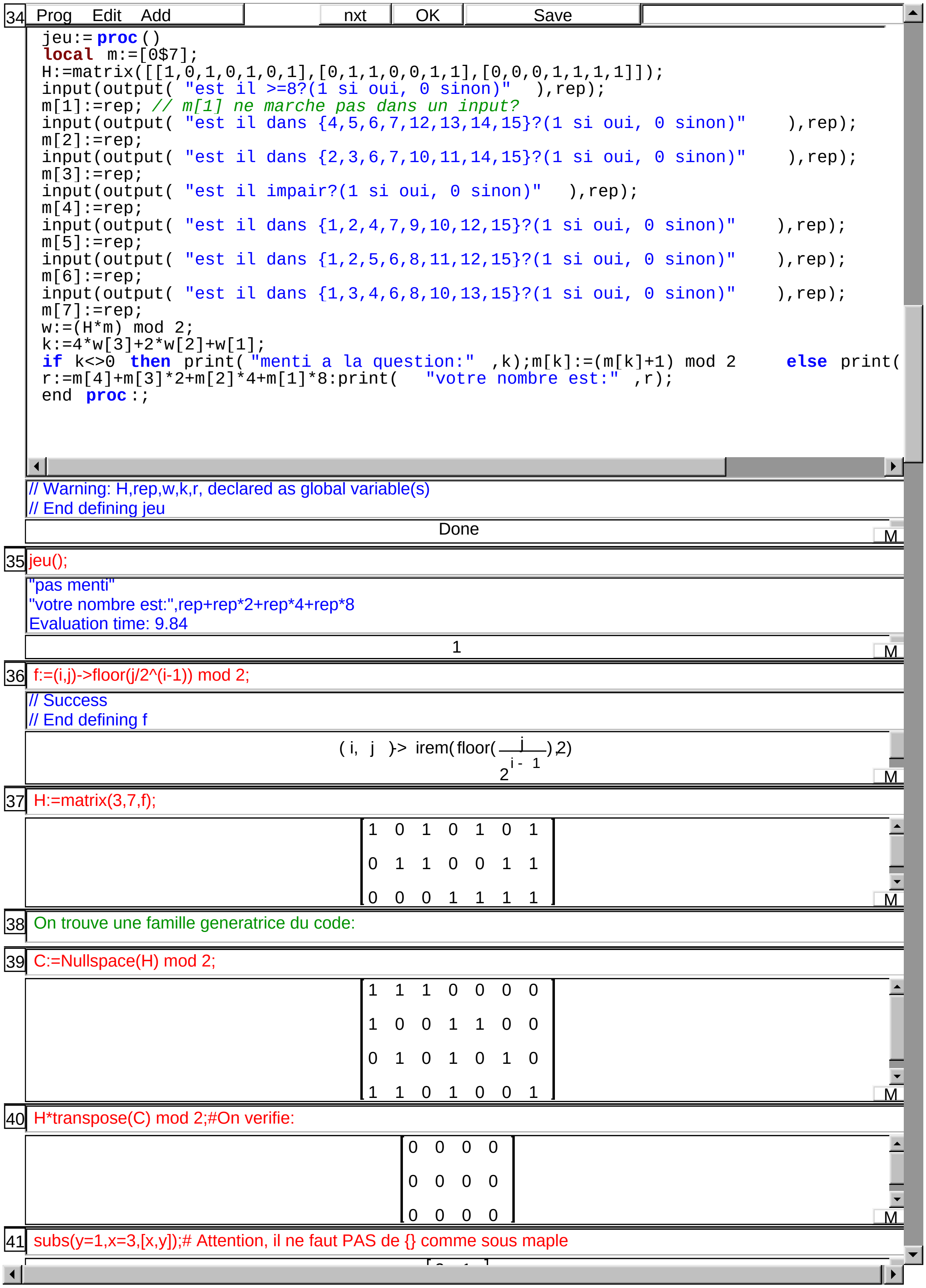
```
Done
```

32
```
r:=m[4]+m[3]*2+m[2]*4+m[1]*8:print("votre nombre est:",r);
```

```
"votre nombre est:",13
```

```
( Done,  1  )
```

33 le meme dans une procedure: 1 pour oui a la question



34

```
jeu:=proc()
local m:=[0$7];
H:=matrix([[1,0,1,0,1,0,1],[0,1,1,0,0,1,1],[0,0,0,1,1,1,1]]);
input(output( "est il >=8?(1 si oui, 0 sinon)"  ),rep);
m[1]:=rep; // m[1] ne marche pas dans un input?
input(output( "est il dans {4,5,6,7,12,13,14,15}?(1 si oui, 0 sinon)"   ),rep);
m[2]:=rep;
input(output( "est il dans {2,3,6,7,10,11,14,15}?(1 si oui, 0 sinon)"   ),rep);
m[3]:=rep;
input(output( "est il impair?(1 si oui, 0 sinon)"  ),rep);
m[4]:=rep;
input(output( "est il dans {1,2,4,7,9,10,12,15}?(1 si oui, 0 sinon)"   ),rep);
m[5]:=rep;
input(output( "est il dans {1,2,5,6,8,11,12,15}?(1 si oui, 0 sinon)"   ),rep);
m[6]:=rep;
input(output( "est il dans {1,3,4,6,8,10,13,15}?(1 si oui, 0 sinon)"   ),rep);
m[7]:=rep;
w:=(H*m) mod 2;
k:=4*w[3]+2*w[2]+w[1];
if k<>0 then print( "menti a la question:" ,k);m[k]:=(m[k]+1) mod 2   else print(
r:=m[4]+m[3]*2+m[2]*4+m[1]*8:print(  "votre nombre est:" ,r);
end proc:;
```

```
// Warning: H,rep,w,k,r, declared as global variable(s)
// End defining jeu
```

Done

35 jeu();

```
"pas menti"
"votre nombre est:",rep+rep*2+rep*4+rep*8
Evaluation time: 9.84
```

1

36 f:=(i,j)->floor(j/2^(i-1)) mod 2;

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

$$(i,\ j) \to \mathrm{irem}\left(\mathrm{floor}\left(\frac{j}{2^{i-1}}\right),2\right)$$

37 H:=matrix(3,7,f);

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

38 On trouve une famille generatrice du code:

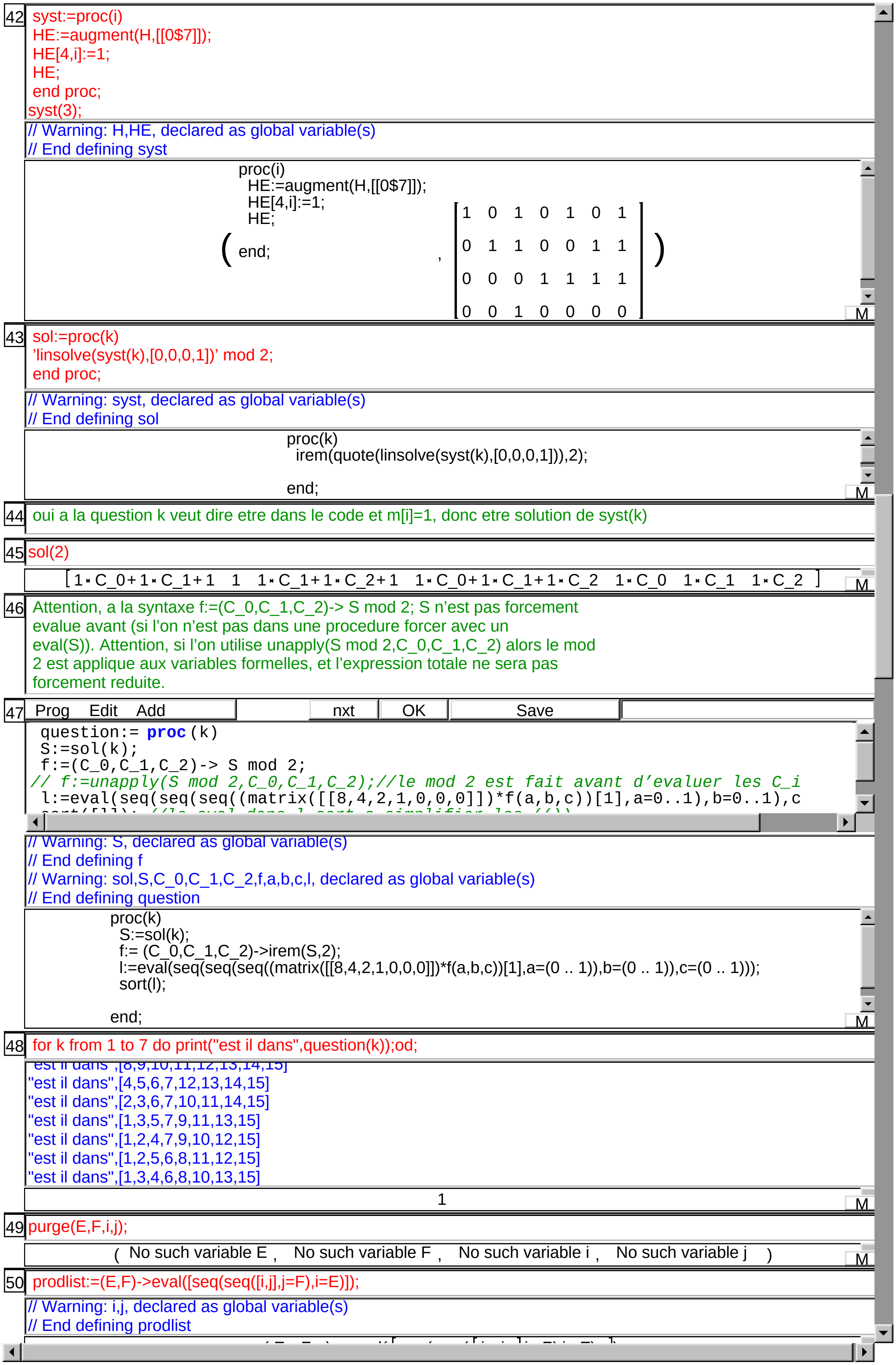
39 C:=Nullspace(H) mod 2;

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

40 H*transpose(C) mod 2;#On verifie:

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

41 subs(y=1,x=3,[x,y]);# Attention, il ne faut PAS de {} comme sous maple



42
```
syst:=proc(i)
 HE:=augment(H,[[0$7]]);
 HE[4,i]:=1;
 HE;
 end proc;
syst(3);
```

```
// Warning: H,HE, declared as global variable(s)
// End defining syst
```

```
proc(i)
  HE:=augment(H,[[0$7]]);
  HE[4,i]:=1;
  HE;
end;
```

$$\left(\text{proc(i) ... end;} , \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}\right)$$

43
```
sol:=proc(k)
 'linsolve(syst(k),[0,0,0,1])' mod 2;
 end proc;
```

```
// Warning: syst, declared as global variable(s)
// End defining sol
```

```
proc(k)
  irem(quote(linsolve(syst(k),[0,0,0,1])),2);

end;
```

44 oui a la question k veut dire etre dans le code et m[i]=1, donc etre solution de syst(k)

45
```
sol(2)
```

```
[1*C_0+1*C_1+1  1  1*C_1+1*C_2+1  1*C_0+1*C_1+1*C_2  1*C_0  1*C_1  1*C_2 ]
```

46 Attention, a la syntaxe f:=(C_0,C_1,C_2)-> S mod 2; S n'est pas forcement evalue avant (si l'on n'est pas dans une procedure forcer avec un eval(S)). Attention, si l'on utilise unapply(S mod 2,C_0,C_1,C_2) alors le mod 2 est applique aux variables formelles, et l'expression totale ne sera pas forcement reduite.

47

```
 question:= proc(k)
 S:=sol(k);
 f:=(C_0,C_1,C_2)-> S mod 2;
// f:=unapply(S mod 2,C_0,C_1,C_2);//le mod 2 est fait avant d'evaluer les C_i
 l:=eval(seq(seq(seq((matrix([[8,4,2,1,0,0,0]])*f(a,b,c))[1],a=0..1),b=0..1),c
```

```
// Warning: S, declared as global variable(s)
// End defining f
// Warning: sol,S,C_0,C_1,C_2,f,a,b,c,l, declared as global variable(s)
// End defining question
```

```
proc(k)
  S:=sol(k);
  f:= (C_0,C_1,C_2)->irem(S,2);
  l:=eval(seq(seq(seq((matrix([[8,4,2,1,0,0,0]])*f(a,b,c))[1],a=(0 .. 1)),b=(0 .. 1)),c=(0 .. 1)));
  sort(l);

end;
```

48
```
for k from 1 to 7 do print("est il dans",question(k));od;
```

```
"est il dans",[8,9,10,11,12,13,14,15]
"est il dans",[4,5,6,7,12,13,14,15]
"est il dans",[2,3,6,7,10,11,14,15]
"est il dans",[1,3,5,7,9,11,13,15]
"est il dans",[1,2,4,7,9,10,12,15]
"est il dans",[1,2,5,6,8,11,12,15]
"est il dans",[1,3,4,6,8,10,13,15]
```

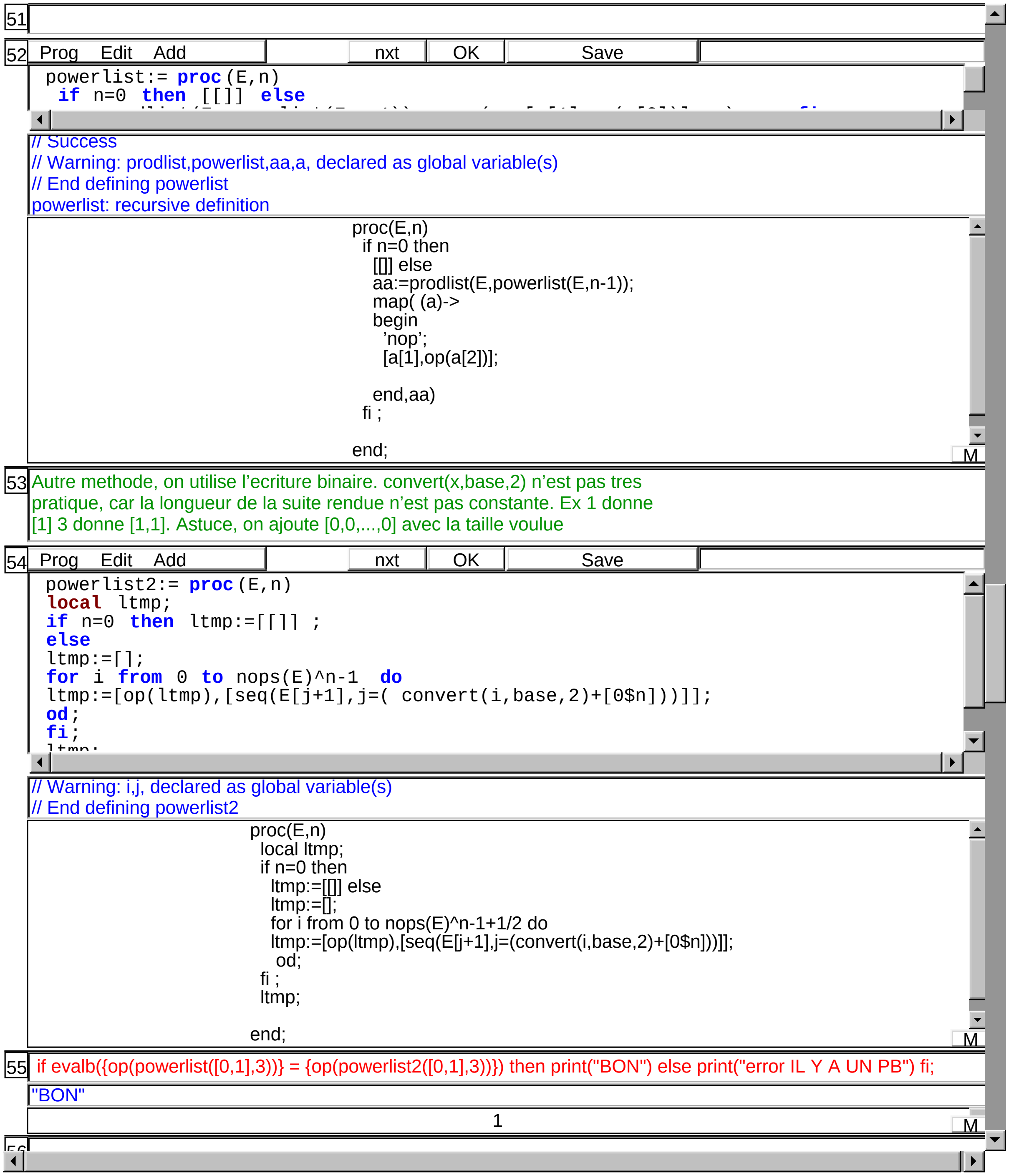
```
1
```

49
```
purge(E,F,i,j);
```

```
( No such variable E ,  No such variable F ,  No such variable i ,  No such variable j  )
```

50
```
prodlist:=(E,F)->eval([seq(seq([i,j],j=F),i=E)]);
```

```
// Warning: i,j, declared as global variable(s)
// End defining prodlist
```



```
powerlist:= proc(E,n)
  if n=0 then [[]] else
```

```
// Success
// Warning: prodlist,powerlist,aa,a, declared as global variable(s)
// End defining powerlist
powerlist: recursive definition
```

```
proc(E,n)
  if n=0 then
    [[]] else
    aa:=prodlist(E,powerlist(E,n-1));
    map( (a)->
    begin
      'nop';
      [a[1],op(a[2])];

    end,aa)
  fi ;

end;
```

Autre methode, on utilise l'ecriture binaire. convert(x,base,2) n'est pas tres pratique, car la longueur de la suite rendue n'est pas constante. Ex 1 donne [1] 3 donne [1,1]. Astuce, on ajoute [0,0,...,0] avec la taille voulue

```
powerlist2:= proc(E,n)
local ltmp;
if n=0 then ltmp:=[[]] ;
else
ltmp:=[];
for i from 0 to nops(E)^n-1  do
ltmp:=[op(ltmp),[seq(E[j+1],j=( convert(i,base,2)+[0$n]))]];
od;
fi;
```

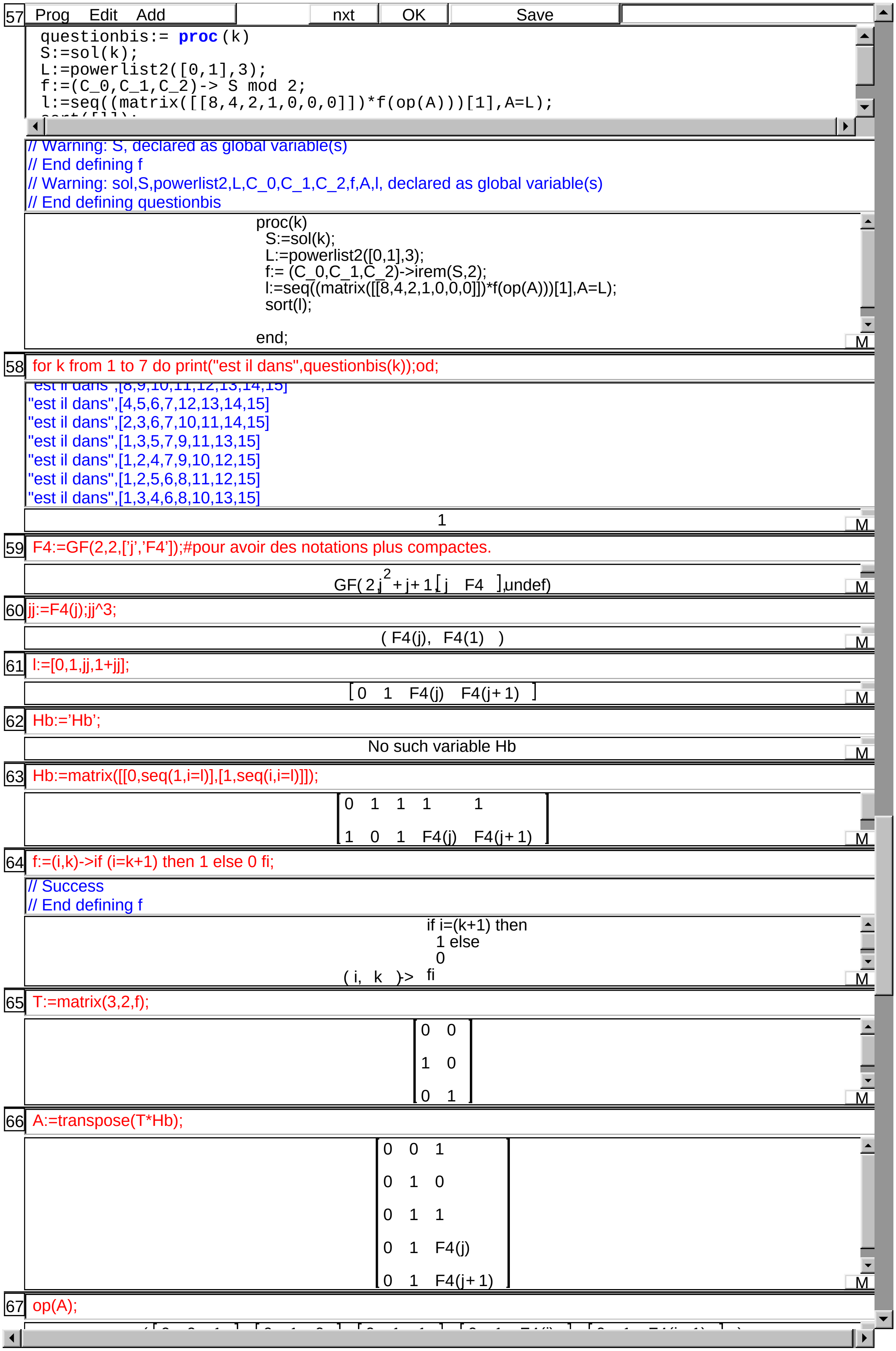
```
// Warning: i,j, declared as global variable(s)
// End defining powerlist2
```

```
proc(E,n)
  local ltmp;
  if n=0 then
    ltmp:=[[]] else
    ltmp:=[];
    for i from 0 to nops(E)^n-1+1/2 do
    ltmp:=[op(ltmp),[seq(E[j+1],j=(convert(i,base,2)+[0$n]))]];
     od;
  fi ;
  ltmp;

end;
```

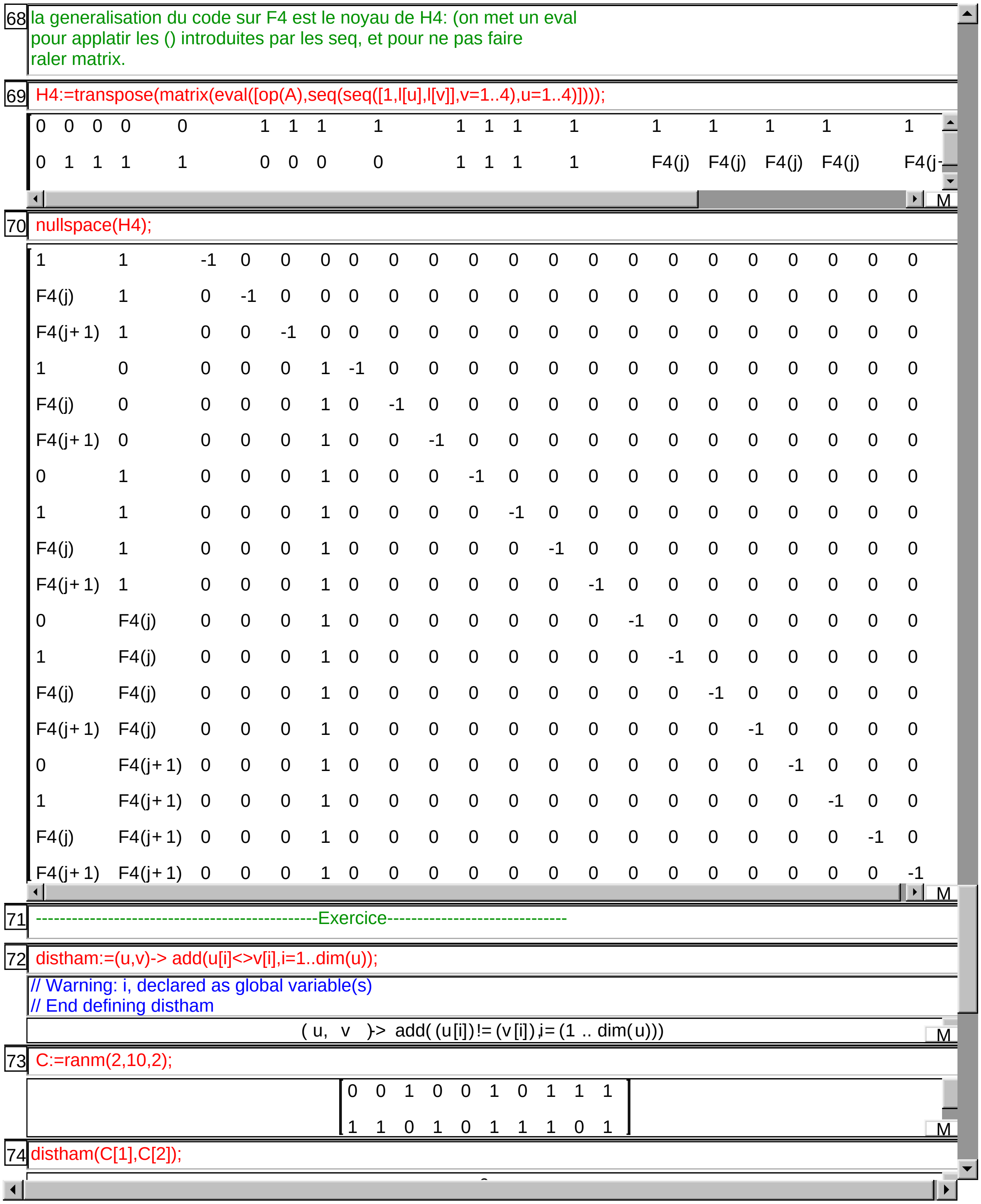
```
if evalb({op(powerlist([0,1],3))} = {op(powerlist2([0,1],3))}) then print("BON") else print("error IL Y A UN PB") fi;
```

```
"BON"
```

```
1
```



```
57 questionbis:= proc(k)
   S:=sol(k);
   L:=powerlist2([0,1],3);
   f:=(C_0,C_1,C_2)-> S mod 2;
   l:=seq((matrix([[8,4,2,1,0,0,0]])*f(op(A)))[1],A=L);
   sort([l]);

// Warning: S, declared as global variable(s)
// End defining f
// Warning: sol,S,powerlist2,L,C_0,C_1,C_2,f,A,l, declared as global variable(s)
// End defining questionbis

                         proc(k)
                           S:=sol(k);
                           L:=powerlist2([0,1],3);
                           f:= (C_0,C_1,C_2)->irem(S,2);
                           l:=seq((matrix([[8,4,2,1,0,0,0]])*f(op(A)))[1],A=L);
                           sort(l);

                         end;

58 for k from 1 to 7 do print("est il dans",questionbis(k));od;

"est il dans",[8,9,10,11,12,13,14,15]
"est il dans",[4,5,6,7,12,13,14,15]
"est il dans",[2,3,6,7,10,11,14,15]
"est il dans",[1,3,5,7,9,11,13,15]
"est il dans",[1,2,4,7,9,10,12,15]
"est il dans",[1,2,5,6,8,11,12,15]
"est il dans",[1,3,4,6,8,10,13,15]

                                   1

59 F4:=GF(2,2,['j','F4']);#pour avoir des notations plus compactes.

                         GF( 2,j^2 + j+ 1,[j   F4  ],undef)

60 jj:=F4(j);jj^3;

                              ( F4(j),  F4(1)   )

61 l:=[0,1,jj,1+jj];

                              [0   1   F4(j)   F4(j+ 1) ]

62 Hb:='Hb';

                              No such variable Hb

63 Hb:=matrix([[0,seq(1,i=l)],[1,seq(i,i=l)]]);

                            [0   1   1   1        1     ]
                            [1   0   1   F4(j)   F4(j+ 1)]

64 f:=(i,k)->if (i=k+1) then 1 else 0 fi;

// Success
// End defining f

                                   if i=(k+1) then
                                     1 else
                                     0
                            ( i,  k  )->  fi

65 T:=matrix(3,2,f);

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

66 A:=transpose(T*Hb);

                               [0   0   1       ]
                               [0   1   0       ]
                               [0   1   1       ]
                               [0   1   F4(j)   ]
                               [0   1   F4(j+ 1)]

67 op(A);
```



68 la generalisation du code sur F4 est le noyau de H4: (on met un eval
pour applatir les () introduites par les seq, et pour ne pas faire
raler matrix.

69
```
H4:=transpose(matrix(eval([op(A),seq(seq([1,l[u],l[v]],v=1..4),u=1..4)])));
```
```
0  0  0  0     0        1  1  1     1        1  1  1     1        1      1      1      1        1

0  1  1  1     1        0  0  0     0        1  1  1     1        F4(j)  F4(j)  F4(j)  F4(j)    F4(j-
```

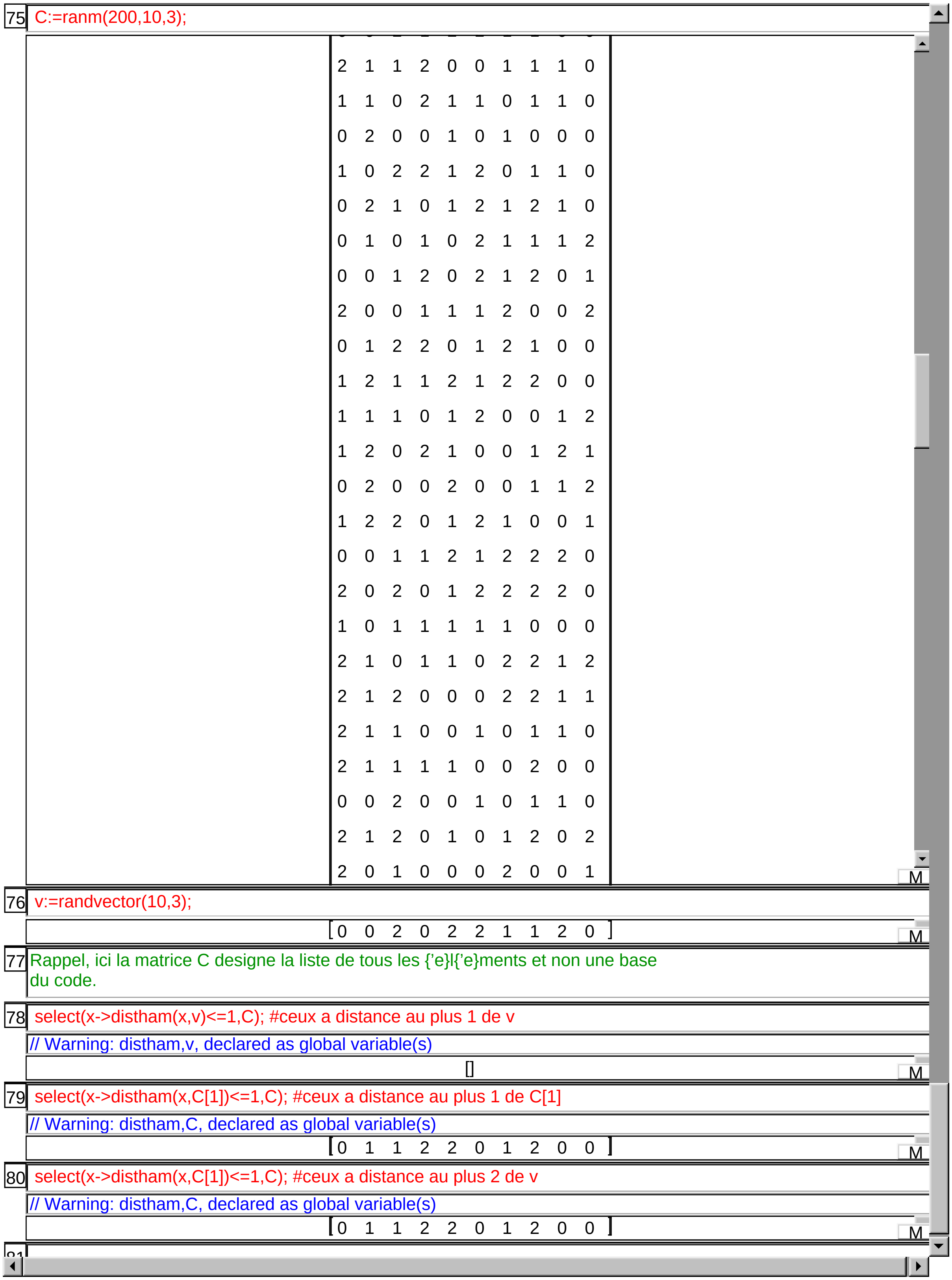
70
```
nullspace(H4);
```
```
1         1         -1   0    0    0  0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
F4(j)     1         0    -1   0    0  0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
F4(j+1)   1         0    0    -1   0  0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
1         0         0    0    0    1  -1   0    0    0    0    0    0    0    0    0    0    0    0    0    0
F4(j)     0         0    0    0    1  0    -1   0    0    0    0    0    0    0    0    0    0    0    0    0
F4(j+1)   0         0    0    0    1  0    0    -1   0    0    0    0    0    0    0    0    0    0    0    0
0         1         0    0    0    1  0    0    0    -1   0    0    0    0    0    0    0    0    0    0    0
1         1         0    0    0    1  0    0    0    0    -1   0    0    0    0    0    0    0    0    0    0
F4(j)     1         0    0    0    1  0    0    0    0    0    -1   0    0    0    0    0    0    0    0    0
F4(j+1)   1         0    0    0    1  0    0    0    0    0    0    -1   0    0    0    0    0    0    0    0
0         F4(j)     0    0    0    1  0    0    0    0    0    0    0    -1   0    0    0    0    0    0    0
1         F4(j)     0    0    0    1  0    0    0    0    0    0    0    0    -1   0    0    0    0    0    0
F4(j)     F4(j)     0    0    0    1  0    0    0    0    0    0    0    0    0    -1   0    0    0    0    0
F4(j+1)   F4(j)     0    0    0    1  0    0    0    0    0    0    0    0    0    0    -1   0    0    0    0
0         F4(j+1)   0    0    0    1  0    0    0    0    0    0    0    0    0    0    0    -1   0    0    0
1         F4(j+1)   0    0    0    1  0    0    0    0    0    0    0    0    0    0    0    0    -1   0    0
F4(j)     F4(j+1)   0    0    0    1  0    0    0    0    0    0    0    0    0    0    0    0    0    -1   0
F4(j+1)   F4(j+1)   0    0    0    1  0    0    0    0    0    0    0    0    0    0    0    0    0    0    -1
```

71 --Exercice-----------------------------

72
```
distham:=(u,v)-> add(u[i]<>v[i],i=1..dim(u));
```
```
// Warning: i, declared as global variable(s)
// End defining distham
```
```
(u, v )-> add( (u[i])!= (v[i]) ,i= (1 .. dim(u)))
```

73
```
C:=ranm(2,10,2);
```
```
[ 0  0  1  0  0  1  0  1  1  1 ]
[ 1  1  0  1  0  1  1  1  0  1 ]
```

74
```
distham(C[1],C[2]);
```



75 `C:=ranm(200,10,3);`

```
2  1  1  2  0  0  1  1  1  0
1  1  0  2  1  1  0  1  1  0
0  2  0  0  1  0  1  0  0  0
1  0  2  2  1  2  0  1  1  0
0  2  1  0  1  2  1  2  1  0
0  1  0  1  0  2  1  1  1  2
0  0  1  2  0  2  1  2  0  1
2  0  0  1  1  1  2  0  0  2
0  1  2  2  0  1  2  1  0  0
1  2  1  1  2  1  2  2  0  0
1  1  1  0  1  2  0  0  1  2
1  2  0  2  1  0  0  1  2  1
0  2  0  0  2  0  0  1  1  2
1  2  2  0  1  2  1  0  0  1
0  0  1  1  2  1  2  2  2  0
2  0  2  0  1  2  2  2  2  0
1  0  1  1  1  1  1  0  0  0
2  1  0  1  1  0  2  2  1  2
2  1  2  0  0  0  2  2  1  1
2  1  1  0  0  1  0  1  1  0
2  1  1  1  1  0  0  2  0  0
0  0  2  0  0  1  0  1  1  0
2  1  2  0  1  0  1  2  0  2
2  0  1  0  0  0  2  0  0  1
```

76 `v:=randvector(10,3);`

```
[0  0  2  0  2  2  1  1  2  0]
```

77 Rappel, ici la matrice C designe la liste de tous les {'e}l{'e}ments et non une base du code.

78 `select(x->distham(x,v)<=1,C); #ceux a distance au plus 1 de v`

```
// Warning: distham,v, declared as global variable(s)
[]
```

79 `select(x->distham(x,C[1])<=1,C); #ceux a distance au plus 1 de C[1]`

```
// Warning: distham,C, declared as global variable(s)
[0  1  1  2  2  0  1  2  0  0]
```

80 `select(x->distham(x,C[1])<=1,C); #ceux a distance au plus 2 de v`

```
// Warning: distham,C, declared as global variable(s)
[0  1  1  2  2  0  1  2  0  0]
```