

```

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 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
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

```

$$\begin{bmatrix} 4 & 2 \\ 3 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 2 \\ 3 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 5 \\ 3 & 5 \end{bmatrix}$$

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

$$1$$

$$1 \cdot x + 1$$

$$1 \cdot x + 1$$

$$[0]$$

$$[0]$$

$$[C_0]$$

$$2 \cdot x + y$$

$$(x, y) \rightarrow y$$

$$2$$

$$(x, y) \rightarrow \text{irem}(S, 2)$$

$$y$$

$$(x, y) \rightarrow \text{irem}(\text{eval}(S), 2)$$

```

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: S x y 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

```



39 C:=Nullspace(H) mod 2;

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

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

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

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

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

```
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; ,  $\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$  )
```

```
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 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.

46 Prog Edit Add

```
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=0..1);
sort(l); //le eval dans l sert a simplifier les (())
end proc ;
```

// Warning: S sol 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);
```



```

56 jj:=F4(j);jj^3;
      ( F4(j), F4(1) )
57 l:=[0,1,jj,1+jj];
      [ 0 1 F4(j) F4(j+1) ]
58 Hb:='Hb';
      Hb not assigned
59 Hb:=matrix([[0,seq(1,i=1)],[1,seq(i,i=1)]]);
      [ 0 1 1 1 1
        1 0 1 F4(j) F4(j+1) ]
60 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
61 T:=matrix(3,2,f);
      [ 0 0
        1 0
        0 1 ]
62 A:=transpose(T*Hb);
      [ 0 0 1
        0 1 0
        0 1 1
        0 1 F4(j)
        0 1 F4(j+1) ]
63 op(A);
      ([ 0 0 1 ], [ 0 1 0 ], [ 0 1 1 ], [ 0 1 F4(j) ], [ 0 1 F4(j+1) ] )
64 la generalisation du code sur F4 est le noyau H4: (on met un eval
pour applatir les () introduites par les seq, et pour ne pas faire
raler matrix.
65 H4:=transpose(matrix(eval([op(A),seq(seq([1,[u],[v]],v=1..4),u=1..4)])));
      0 0 0 0 0 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+
      1 0 1 F4(j) F4(j+1) 0 1 F4(j) F4(j+1) 0 1 F4(j) F4(j+1) 0 1 F4(j) F4(j+1) 0

```

```

66 nullspace(H4);
1      1      -1  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
F4(j)  0      0  0  0  1  0 -1  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      1      0  0  0  1  0  0  0 -1  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
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

```

-----Exercice-----

```

68 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)))

```

```

69 C:=ranm(2,10,2);
| 1 1 0 0 1 1 1 1 1 1 |
| 0 0 1 1 1 0 0 1 0 1 |

```

```

70 distham(C[1],C[2]);
7

```

```

71 C:=ranm(200,10,3);
| 2 0 0 1 1 0 2 0 2 0 |
| 1 1 2 1 2 1 2 1 0 0 |
| 2 0 1 1 2 2 0 0 1 0 |
| 0 0 1 2 0 2 0 1 0 2 |
| 1 0 1 0 1 0 0 0 1 1 |
| 2 1 1 0 1 0 1 0 0 0 |
| 2 0 1 0 0 2 2 0 0 0 |
| 2 2 2 2 2 0 2 2 1 0 |
| 0 2 2 1 2 2 0 2 0 0 |
| 0 2 1 1 1 0 2 2 0 1 |
| 0 0 1 2 0 1 2 1 1 0 |
| 0 2 1 2 2 1 2 0 1 1 |
| 2 1 0 0 2 1 2 2 1 1 |

```

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

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

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

```
73 Rappel, ici la matrice C designe la liste de tous les 'el'ements et non une base du code.
```

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

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

```
[]
```

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

```
// Warning: C distham declared as global variable(s)
```

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

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

```
// Warning: C distham declared as global variable(s)
```

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