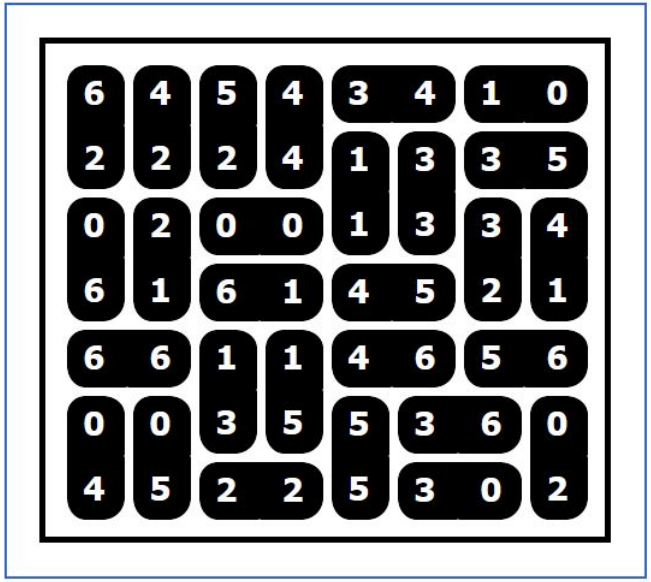
Projet Prolog :
Solveur de Dominosa

- Aguesse Nathan

Dominosa ?

6	4	5	4	3	4	1	0
2	2	2	4	1	3	3	5
0	2	0	0	1	3	3	4
6	1	6	1	4	5	2	1
6	6	1	1	4	6	5	6
0	0	3	5	5	3	6	0
4	5	2	2	5	3	0	2

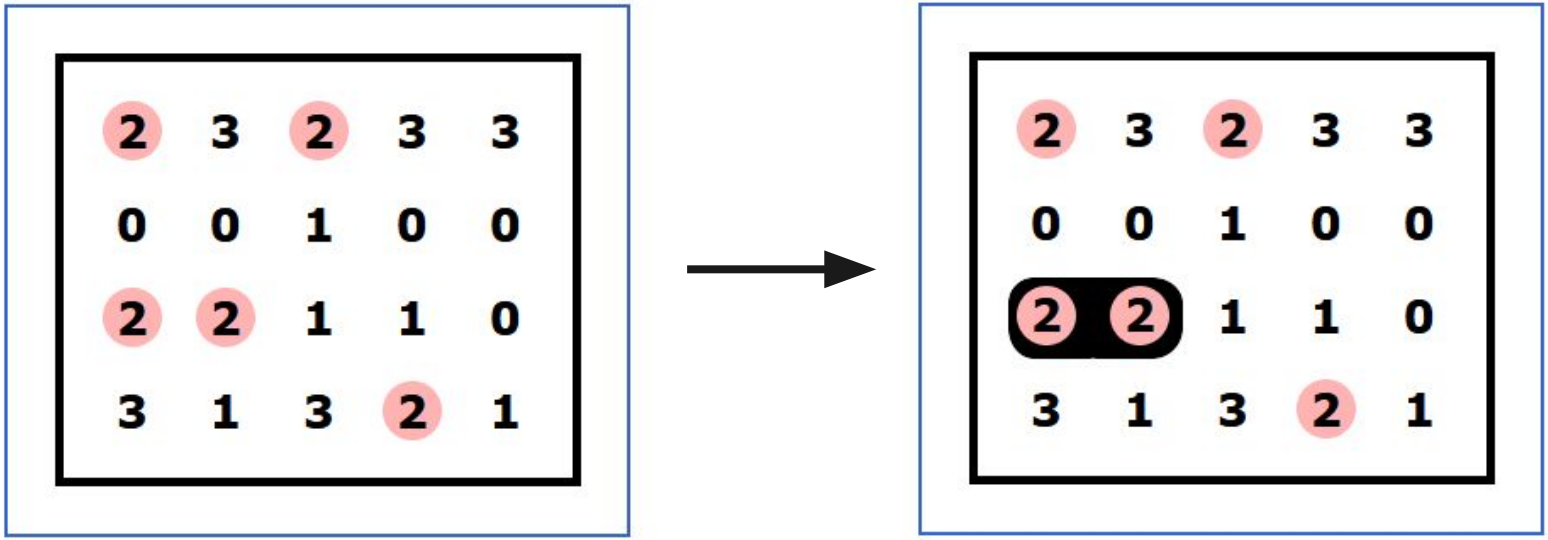


Comment jouer ?

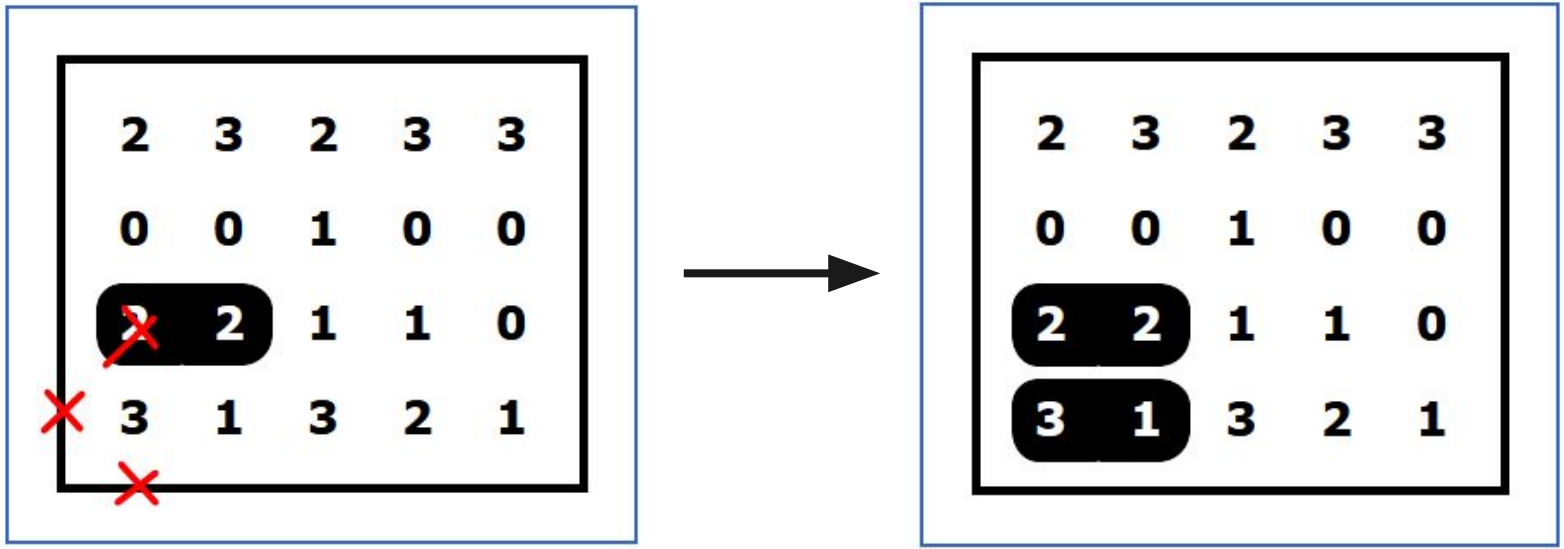
On peut pas vraiment poser des dominos n’importe comment, de la même manière d’un sudoku, il ne faut pas qu’il y ait 2 même domino posé dans la grille ce qui n’est pas valide.

Ainsi, pour reconnaître les coups que l’on peut faire, on a droit à 3 cas différents :

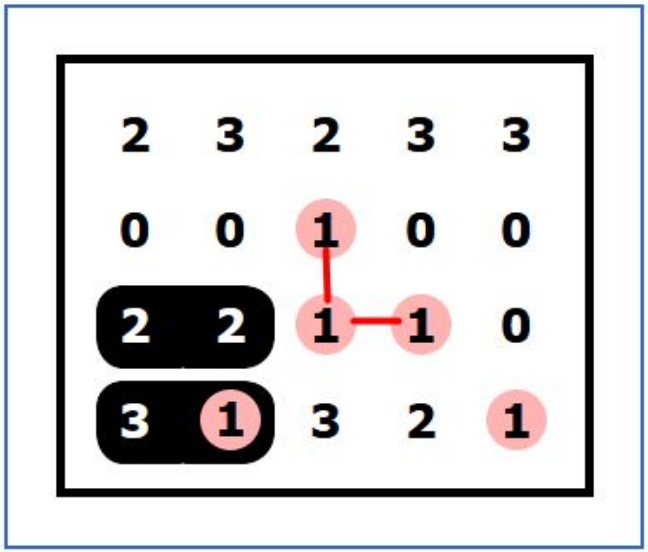
1er cas :

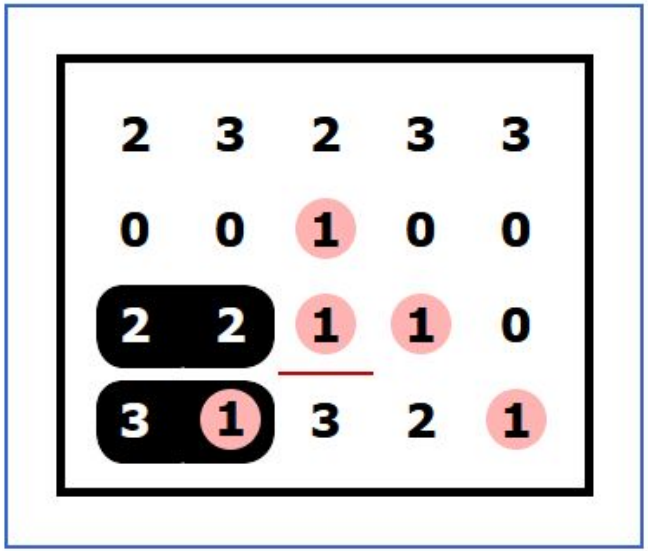


2ème cas :



3ème cas :



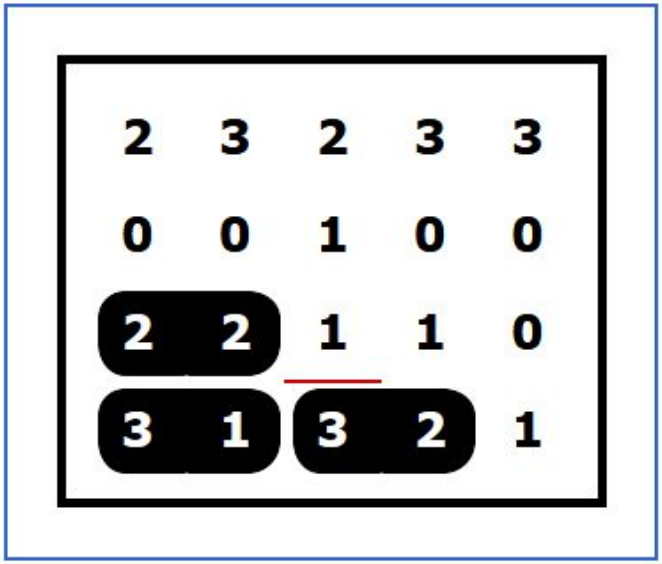


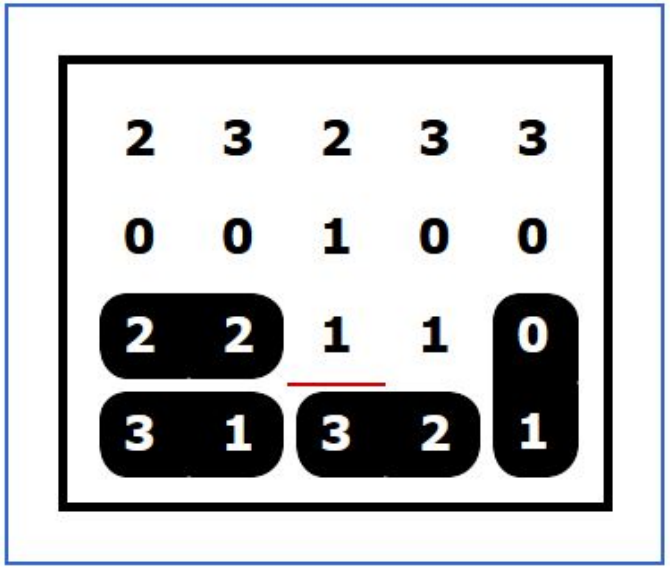
Etc…

2	3	2	3	3
0	0	1	0	0
2	2	1	1	0
3	1	3	2	1

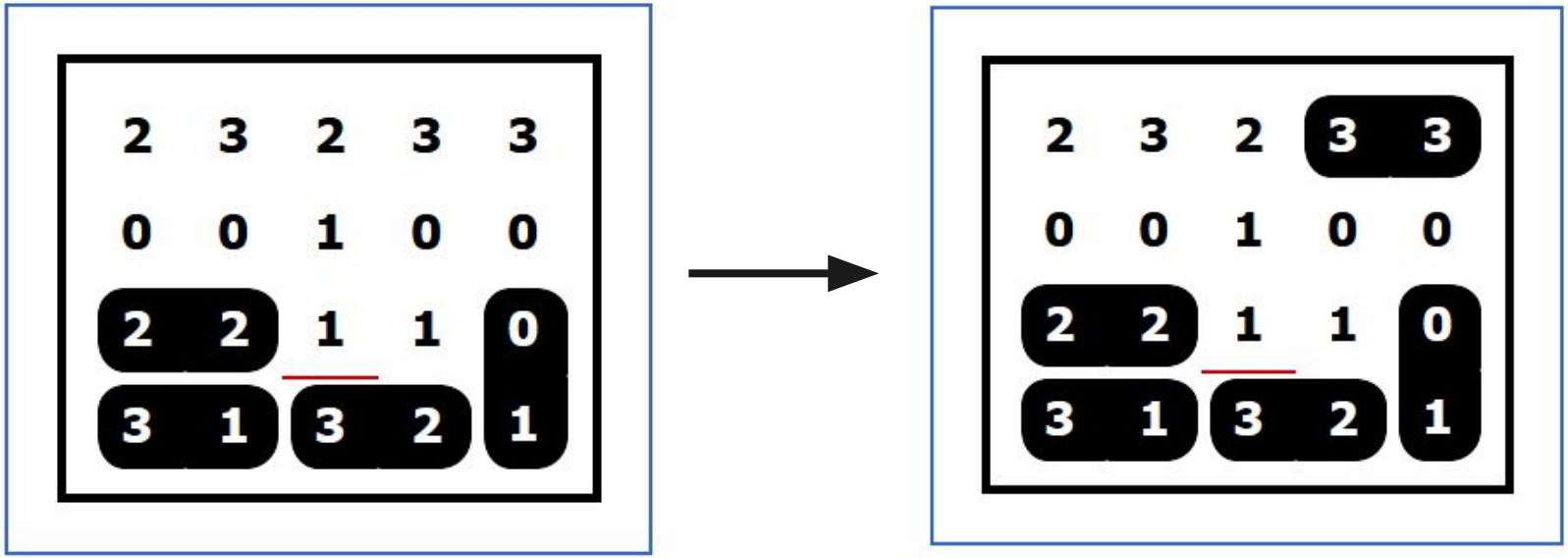
2	3	2	3	3
0	0	1	0	0
2	2	1	1	0
3	1	3	2	1

Etc…

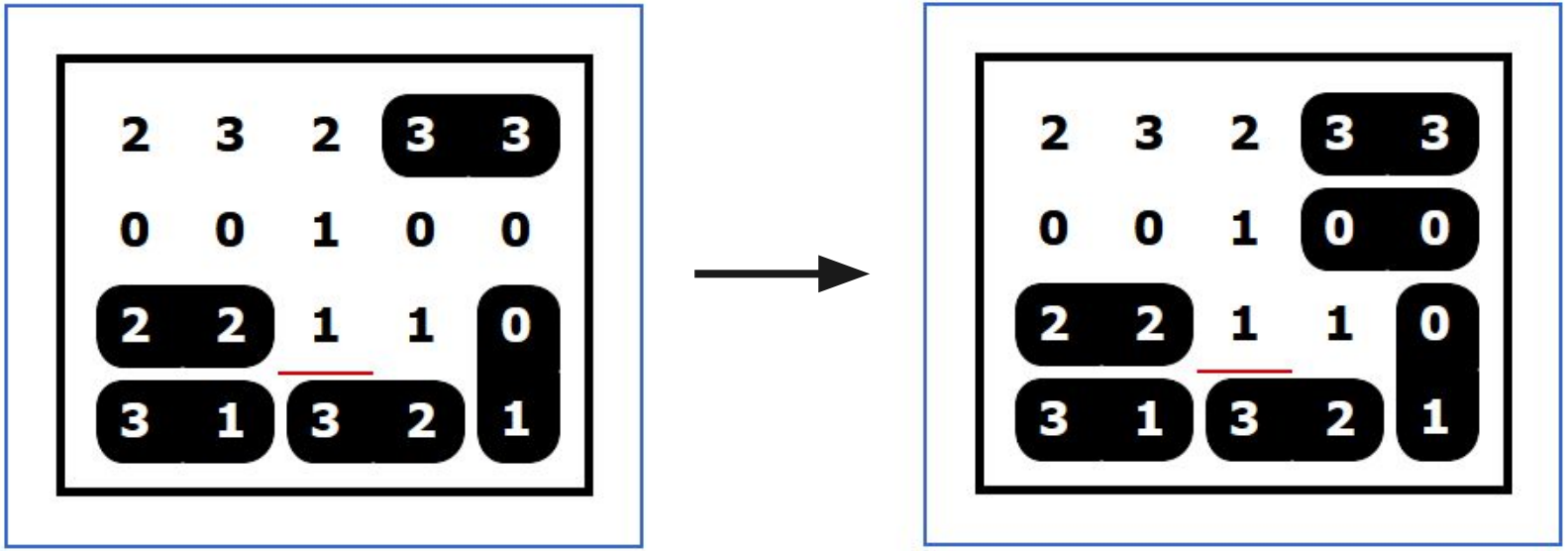




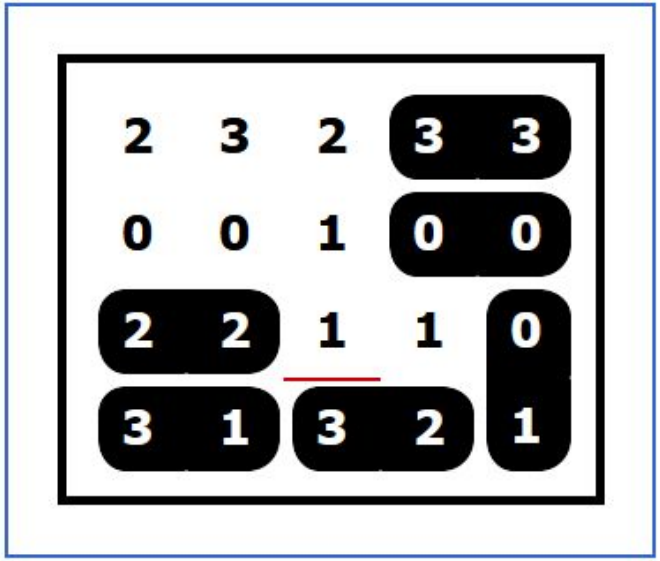
Etc…

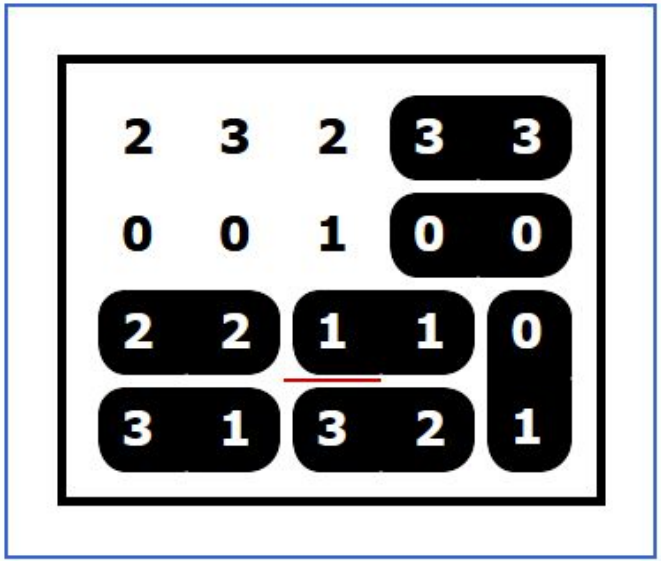


Etc…

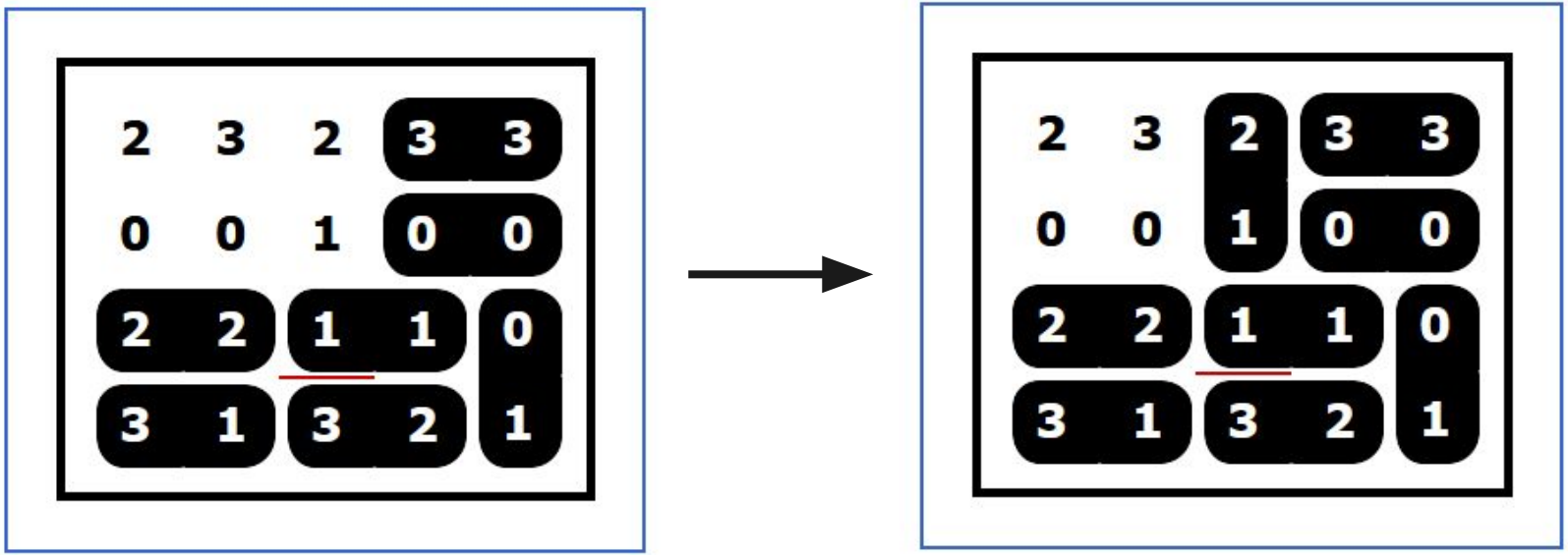


Etc…

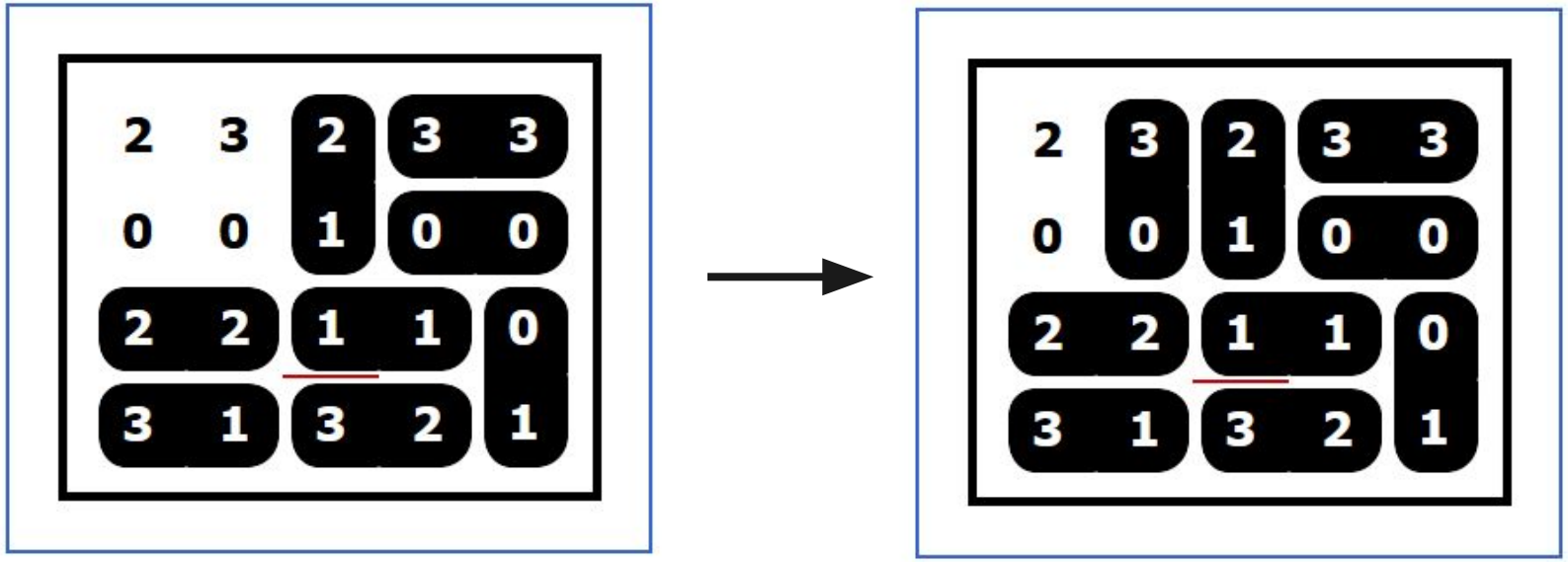




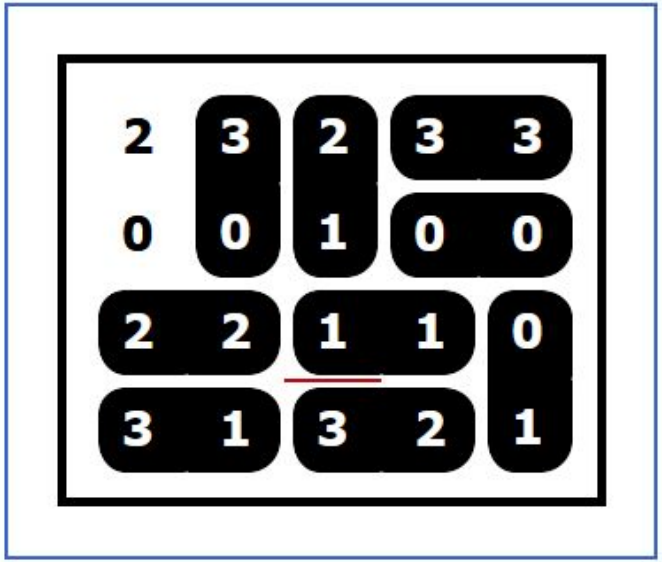
Etc…

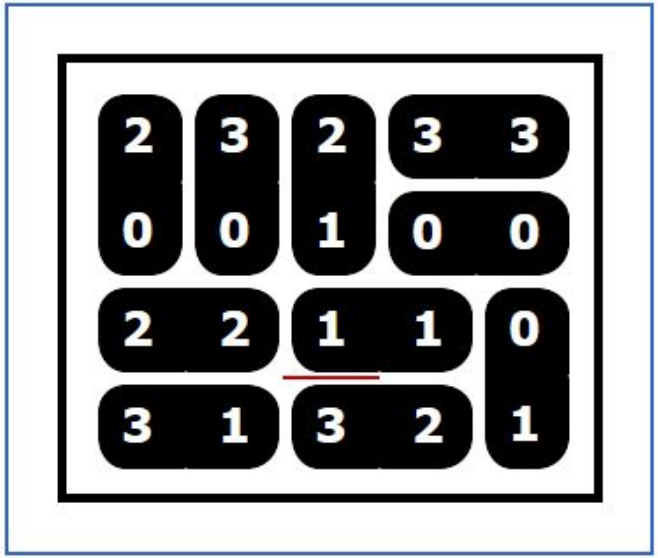


Etc…



Fin





Structure du code :

Une vingtaine de prédicats de tout genre, permettant d'avoir tous les "outils" dont je vais avoir besoin lors du fonctionnement du programme :

rows(M,R)	possibilities(M,L)	domino_id(D,LD,I,LI)	sup(I,D,LI,LD,NLI,NLD)
columns(M,C)	clear(M,L)	change_m(M,M1,I)	sup1(I,LI,LD,NLI,NLD)
create_list(N,L)	poss(M,L)	add_m(M,M1,I,Car,N)	check_same(SD,LD,LI,N,L)
check_element(M,L	id(A,B,R,C,N,L)	distinct_id(I,M,L)	group(SD,LD,LI,G)
dominos(Spe,Spe,L)	distinct(D,M,L)	nmb_id(I,M,N)	commun(G,I)
sup_domino(L,NL,D)	nmb(D,M,N)	find_id(I,D,L)	sup_spe(I,D,LI,LD,NLI,NLD)

Structure du code : Solution

```
solution(M,S):-
    rows(M,R),
    columns(M,C),
    RC is R*C,
    create_list(RC,ListeID),
    check_element(M,ListeN),
    dominos(ListeN,ListeN,SetDomino),
    poss(M,LD),
    id(0,0,R,C,0,LI),
    dominosa(M,S,R,C,LD,LI,ListeID,SetDomino,SetDomino).
```

Structure du code : Dominosa cas N°1

```
dominosa(M,S,R,C,LD,LI,ListeID,SetDomino,SetDominoSPE):-
    distinct(SetDomino,LD,D),
    domino_id(D,LD,L,LI),
    change_m(M,M1,L),
    sup_domino(SetDomino,NSetDomino,D),
    sup_domino(SetDominoSPE,NSetDominoSPE,D),
    sup(L,D,LI,LD,NLI,NLD),
    dominosa(M1,S,R,C,NLD,NLI,ListeID,NSetDomino,NSetDominoSPE),!.
```

Structure du code : Dominosa cas N°2

```
dominosa(M,S,R,C,LD,LI,ListeID,SetDomino,SetDominoSPE):-
    distinct_id(ListeID,LI,L),
    change_m(M,M1,L),
    domino_id(D,LD,L,LI),
    sup_domino(SetDomino,NSetDomino,D),
    sup_domino(SetDominoSPE,NSetDominoSPE,D),
    sup(L,D,LI,LD,NLI,NLD),
    dominosa(M1,S,R,C,NLD,NLI,ListeID,NSetDomino,NSetDominoSPE),!.
```

Structure du code : Dominosa cas N°3

```
dominosa(M,S,R,C,LD,LI,ListeID,SetDomino,SetDominoSPE):-
    check_same(SetDominoSPE,LD,LI,N,L),
    sup_spe(N,L,LI,LD,NLI,NLD),
    sup_domino(SetDominoSPE,NSetDominoSPE,L),
    dominosa(M,S,R,C,NLD,NLI,ListeID,SetDomino,NSetDominoSPE),!.
```

Structure du code : Dominosa cas de fin.

```
dominosa(S,S,_,_,[],[],_,[],_):-!.
```

Résultat :

```
?- solution([[6,4,5,4,3,4,1,0],
             [2,2,2,4,1,3,3,5],
             [0,2,0,0,1,3,3,4],
             [6,1,6,1,4,5,2,1],
             [6,6,1,1,4,6,5,6],
             [0,0,3,5,5,3,6,0],
             [4,5,2,2,5,3,0,2]],S).
```

```
S = [['S','S','S','S','E','W','E','W'],
     ['N','N','N','N','S','S','E','W'],
     ['S','S','E','W','N','N','S','S'],
     ['N','N','E','W','E','W','N','N'],
     ['E','W','S','S','E','W','E','W'],
     ['S','S','N','N','S','E','W','S'],
     ['N','N','E','W','N','E','W','N']].
```