

# Compte Rendu TP2

## Question

```
:-lib(ic).
:-lib(ic_symbolic).

:-local domain(pays(anglais,espagnol,ukrainien,norvegien,japonais)).
:-local domain(couleur(rouge,verte,jaune,bleue,blanche)).
:-local domain(boisson(cafe,the,eau,lait,jus)).
:-local domain(voiture(bmw,toyota,ford,honda,datsum)).
:-local domain(animal(chien,serpents,zebre,renard,cheval)).
```

## Question2

```
domaines_maison(m(Pays,Couleur,Boisson,Voiture,Animal,_)):-
    Pays&::pays,
    Couleur&::couleur,
    Boisson&::boisson,
    Voiture&::voiture,
    Animal&::animal.
```

## Question3

```
rue([m(P1, C1, B1, V1, A1, 1), m(P2, C2, B2, V2, A2, 2), m(P3, C3, B3, V3, A3, 3), m(P4, C4, B4, V4, A4, 4), m(P5, C5, B5, V5, A5, 5)],
    domaines_maison(m(P1, C1, B1, V1, A1, 1)),
    domaines_maison(m(P2, C2, B2, V2, A2, 2)),
    domaines_maison(m(P3, C3, B3, V3, A3, 3)),
    domaines_maison(m(P4, C4, B4, V4, A4, 4)),
    domaines_maison(m(P5, C5, B5, V5, A5, 5)),
    ic_symbolic:alldifferent([P1, P2, P3, P4, P5]),
    ic_symbolic:alldifferent([C1, C2, C3, C4, C5]),
    ic_symbolic:alldifferent([B1, B2, B3, B4, B5]),
    ic_symbolic:alldifferent([V1, V2, V3, V4, V5]),
    ic_symbolic:alldifferent([A1, A2, A3, A4, A5])).
```

## Tests

```
[m(anglais,rouge,cafe,bmw,chien,1),m(espagnol,verte,the,toyota,serpents,2),m(ukrainien,jaune,eau,ford,zebre,3),m(norvegien,bleue,vin,volvo,renard,4),m(japonais,blanche,whisky,cheval,5)].
```

## Question4

```
ecrit_maisons(Rue):-
    (foreach(Elem,Rue)
    do
        writeln(Elem)
    ).
```

## Question5

```
getVarList(Rue,Liste):-
    (foreach(m(P, C, B, V, A,_),Rue),
    foreach(P,C,B,V,A,_,Liste)
    do
        write(P),write(C),write(B),write(V),write(A),write(_),write(' '),nl
    ).
```

```

fromto([],In,Out,Liste)
do
    Out = [P,C,B,V,A|In]
).
labeling_symbolic(Liste):-
    (foreach(Elem,Liste)
    do
        ic_symbolic:indomain(Elem)

    ).

```

## Question6

```

enigma_unary(Rue):-
    (foreach(m(P, C, B, V, A, N),Rue)
    do
        ( P &= anglais ) => ( C &= rouge ),
        ( P &= espagnol ) => ( A &= chien ),
        ( C &= verte ) => ( B &= cafe ),
        ( P &= ukrainien ) => ( B &= the ),
        ( V &= bmw ) => ( A &= serpents ),
        ( C &= jaune ) => ( V &= toyota ),
        ( B &= lait ) => ( N #= 3 ),
        ( P &= norvegien ) => ( N #= 1 ),
        ( V &= honda ) => ( B &= jus ),
        ( P &= japonais ) => ( V &= datsun )

    ).

enigma_binary(Rue):-
    (foreach(m(P, C, B, V, A, N),Rue)
    do
        ( C &= verte ) => (foreach(m(_, C2, _, _, _, N2),Rue)
            do
                ( C2 &= blanche ) => ( (N2 #= N+1) or (N2 #= N-1) )
            ),
        (A &= renard) => (foreach(m(_,_, _,V2, _, N2),Rue)
            do
                (V2 &= ford) => ((N2 #= N+1) or (N2 #= N-1))
            ),
        (A &= cheval) => (foreach(m(_,_, _,V2, _, N2),Rue)
            do
                (V2 &= toyota)=>((N2 #= N+1) or (N2 #= N-1))
            ),
        (P &= norvegien)=>(foreach(m(_,C2, _,_, _, N2),Rue)
            do
                (C2 &= bleue)=>((N2 #= N+1) or (N2 #= N-1))
            )
    ).

```

## Question7

```

resoudre(Rue):-
    rue(Rue),
    getVarList(Rue,Liste),
    enigma_unary(Rue),
    labeling_symbolic(Liste),
    ecrit_maisons(Rue).

```