

```

/**
TP Listes Prolog
@author Théo CHAPON
@author Hassan EL OMARI ALAOUI
@version Annee scolaire 2014/2015
*/

/**Partie 1**/

/**Question a
A est élément de la liste X
*/
membre(A,[A|_]).
membre(A,[B|X]):- \==(B,A),membre(A,X).
membre(2,[1,2,5]).
Yes (0.00s cpu, solution 1, maybe more)

membre(A,[1,4,7]).
A = 1
A = 4
A = 7
Yes (0.00s cpu, solution 3, maybe more)

membre(2,[1,5]).
No (0.00s cpu)

/**Question b
N est le nombre d'occurrence de A dans la liste X
*/
compte(A,X,N):- cpt(A,X,0,N).
cpt(_,[],N,N).
cpt(A,[A|X],M,N):- K is M + 1,cpt(A,X,K,N).
cpt(A,[B|X],M,N):- \==(B,A),cpt(A,X,M,N).
/**
compte(2,[1,2,5,7,8,2,1,2],N).
N = 3
Yes (0.00s cpu, solution 1, maybe more)

compte(2,[5,4,7,2,1],1).
Yes (0.00s cpu, solution 1, maybe more)

compte(2,[1,2],3).
No (0.00s cpu)

/**Question c
Y est la liste X à l'envers
*/
renverser(X,Y):- renv(X,[],Y).
renv([],Y,Y).
renv([A|X],Z,Y):- renv(X,[A|Z],Y).
/**
renverser([5,8,7,4],Y).
Y = [4,7,8,5]
Yes (0.00s cpu)
renverser([1,2,5,4],[4,5,2,1]).
Yes (0.00s cpu)
renverser([1,2,3],[2,1,3]).
No (0.00s cpu)

/**
/**Question d
X est une liste palindrome
*/
palind(X):- renverser(X,X).

```

```

palind([1,2,2,3,2,2,1]).
Yes (0.00s cpu)
palind([1,2,2,3,2,1]).
No (0.00s cpu)
*/

/**Question e
A est l'élément de rang N dans la liste X
*/
/**mode +,+,-**/
nieme1(0,[A|_],A).
nieme1(N,[_X],A):- M is N - 1,nieme1(M,X,A).
/**
nieme1(2,[1,2,3,4],A).
A = 3
Yes (0.00s cpu, solution 1, maybe more)

/**mode -,+,-**/
nieme2(N,X,A):- compter(0,N,X,A).
compter(N,N,[A|_],A).
compter(M,N,[B|X],A):- \==(B,A),K is M+1,compter(K,N,X,A).
/**
nieme2(N,[1,2,3,4],2).
N = 1
Yes (0.00s cpu, solution 1, maybe more)

/**mode ?,+,-**/
/**nieme2 marche dans les deux cas*/
nieme2(2,[1,2,3,4],A).
A = 3
Yes (0.00s cpu, solution 1, maybe more)
*/

/**Question f
A n'est pas élément de la liste X
*/
/** avec cut **/
hors_del(A,X):- membre(A,X),!,fail.
hors_del(_,_).
hors_del(1,[2,5,4,8]).
Yes (0.00s cpu)
hors_del(1,[1,2,5]).
No (0.00s cpu)
*/
/** sans cut **/
hors_de2(_,[1]).
hors_de2(A,[B|X]):- \==(A,B),hors_de2(A,X).
/**
hors_de2(1,[2,5,4,8]).
Yes (0.00s cpu)

hors_de2(1,[1,2,5]).
No (0.00s cpu)

*/

/**Question g
les éléments de la liste X sont tous différents
*/
tous_diff(X):- diff(X,[1]).
diff([_],_).
diff([A|X],Y):- hors_de2(A,Y),diff(X,[A|Y]).
/**
tous_diff([1,2,5,4]).
Yes (0.00s cpu)

tous_diff([1,2,5,1]).
No (0.00s cpu)
*/

/**Question h

```

```

T est la concaténation des liste X,Y et Z
*/
conc3([A|X],Y,Z,[A|T]):- conc3(X,Y,Z,T).
conc3([],[A|Y],Z,[A|T]):- conc3([],Y,Z,T).
conc3([],[],Z,Z).
/**mode -, -, +, +*/

/*conc3 peut découper T de toutes les façons possibles*/
/**
conc3([1],[1],[2,3],[4],[1,T]).
T = [1, 2, 3, 4]

conc3([1],[1],[2,3],[4],[1,2,4]).
No (0.00s cpu)

conc3([1],[1],[2,3],[4],[1,2,3,4]).
Yes (0.00s cpu, solution 1, maybe more)

conc3(X,Y,Z,[1,2,3,4]).

X = [1, 2, 3, 4]
Y = []
Z = []

X = [1, 2, 3]
Y = [4]
Z = []

X = [1, 2, 3]
Y = []
Z = [4]

X = [1, 2]
Y = [3, 4]
Z = []

X = [1, 2]
Y = [3]
Z = [4]

X = [1, 2]
Y = []
Z = [3, 4]

X = [1]
Y = [2, 3, 4]
Z = []

X = [1]
Y = [2, 3]
Z = [4]

X = [1]
Y = [2]
Z = [3, 4]

X = [1]
Y = []
Z = [2, 3, 4]

X = []
Y = [1, 2, 3, 4]
Z = []

X = []
Y = [1, 2, 3]
Z = [4]

X = []
Y = []
Z = [3, 4]

X = [1]
Y = [1, 2]
Z = [3, 4]

```

```

X = []
Y = [1]
Z = [2, 3, 4]

X = []
Y = []
Z = [1, 2, 3, 4]
Yes (0.00s cpu, solution 15, maybe more)

*/

/**Question i
la liste X débute par la liste Y
*/
debute_par(_, []).
debute_par([A|X],[A|Y]):- debute_par(X,Y).
/**
debute_par([1,2,5],[1]).
Yes (0.00s cpu)

[eclipse 51]: debute_par([1,2,5],Y).
Y = []

Y = [1]

Y = [1, 2]

Y = [1, 2, 5]
Yes (0.00s cpu, solution 4)

debute_par([1,2,5],[2]).
No (0.00s cpu)

*/

/**Question j
la liste Y est sous liste de la liste X
*/
sous_liste(X,Y):- debute_par(X,Y).
sous_liste(_[X],Y):- sous_liste(X,Y).
/**
sous_liste([1,2,5],Y).
Y = []

Y = [1]

Y = [1, 2]

Y = [1, 2, 5]

Y = []

Y = [2]

Y = [2, 5]

Y = []

Y = [5]

Y = []
Yes (0.00s cpu, solution 10)

sous_liste([1,2,3],[2,3]).
Yes (0.00s cpu, solution 1, maybe more)

sous_liste([1,2,3],[1,3]).
No (0.00s cpu)

*/

/**Question k
Y contient tous les éléments de X une seule fois
*/

```

```
elim([],[ ]).
elim([A|X],[A|Y]):- elim(X,Y),hors_de2(A,Y).
/**
elim([1,5,4,4,5,8,7,4,1,3,6,5],Y).
Y = [8, 7, 4, 1, 3, 6, 5]
Yes (0.00s cpu, solution 1, maybe more)
*/

/**Question 1
la liste Y est le résultat du tri par ordre croissant de la liste d'entiers X
*/
tri([],[ ]).
tri([A|X],Y):-tri(X,Z),insérer(A,Z,Y).

insérer(A,[],[A]).
insérer(A,[E|Z],[E|Y]):- A > E,insérer(A,Z,Y).
insérer(A,[E|Z],[A,E|Z]):- A <= E.
/**
tri([2,1,8,5,4,7,9,6],Y).
Y = [1, 2, 4, 5, 6, 7, 8, 9]
Yes (0.00s cpu, solution 1, maybe more)
*/

/**Partie 2**/
/**Question 1*/
/**Question a
Tous les éléments de l'ensemble X sont présents dans l'ensemble Y
*/
inclus([],[ ]).
inclus([A|X],[B|Y]):- inclus(X,Y),membre(A,Y),hors_de2(A,X).
/**
inclus([1,2],[1,2,3]).
Yes (0.00s cpu, solution 1, maybe more)
No (0.00s cpu)

inclus([1,2,4],[1,2,3]).
No (0.00s cpu)

inclus([1,2,4],Y).
Y = [4, 2, 1|_181]

Y = [4, 2, _179, 1|_183]

Y = [4, 2, _179, _181, 1|_185]

Y = [4, 2, _179, _181, _183, 1|_187]

Y = [4, 2, _179, _181, _183, _185, 1|_189]

...

inclus(X,[1,2]).
X = []

X = [1]

X = [2]

X = [2, 1]

X = [1, 2]
Yes (0.00s cpu, solution 5, maybe more) ?

boucle infinie ...

*/
/**
inclus([1,4,7],[1,5,4,7]).
Yes (0.00s cpu, solution 1, maybe more)

inclus([1,4,7],[1,5,7]).
```

```
No (0.00s cpu)

*/

/**Question b
Au moins un élément de l'ensemble X est hors de l'ensemble Y
*/
non_inclus([A|_],Y):-hors_de2(A,Y).
non_inclus([_|X],Y):-non_inclus(X,Y).

/**
non_inclus([1,4,7],[1,5,7]).
Yes (0.00s cpu, solution 1, maybe more) ?

non_inclus([1,4,7],[1,5,4,7]).
No (0.00s cpu)
*/

/**Question b
Z est l'union ensembliste de X et Y
*/
union_ens([],[],[]).

union_ens([A|X],Y,[A|Z]):- union_ens(X,Y,Z),hors_de2(A,Z).
union_ens([A|X],Y,Z):- union_ens(X,Y,Z),membre(A,Z).

union_ens([],[A|Y],[A|Z]):- union_ens([],Y,Z),hors_de2(A,Z).
union_ens([],[A|Y],Z):- union_ens([],Y,Z),membre(A,Z).

/**
union_ens([1,2],[5,1,5,4],Z).
Z = [1, 2, 5]
Yes (0.00s cpu, solution 1, maybe more)

union_ens([1,2],[5,1,5,4],Z).
Z = [2, 1, 5, 4]
Yes (0.00s cpu, solution 1, maybe more)

union_ens([1,2],[5,1,5,4],[2,1,5,4]).
Yes (0.00s cpu, solution 1, maybe more)
*/

/**Question2**/
```