Université Grenoble Alpes
U.F.R. Sciences de l'Homme et de la Société
L2 MIASHS

T.P. Prolog n°3

Exercice 1 : Duplications

a) Ecrire le prédicat `dupli/2` qui crée deux exemplaires de chaque élément d'une liste donnée dans une nouvelle liste. Exemple :

```
?- dupli([a,b,c,c,d],X).
X = [a,a,b,b,c,c,c,c,d,d]
```

b) Ecrire le prédicat `dupli/3` qui crée N exemplaires de chaque élément d'une liste donnée dans une nouvelle liste. Exemple :

```
?- dupli([a,b,c],3,X).
X = [a,a,a,b,b,b,c,c,c]
```

Exercice 2 : Profondeur

Ecrire le prédicat `profondeur/2` qui détermine la profondeur maximale d'une liste, c'est-à-dire le nombre maximal de niveaux d'imbrication.

```
?- profondeur([1,2],R).
R=1
```

```
?- profondeur([[1,2],[3,[[4]],[5]],6],R).
R= 4
```

Exercice 3 :

Ecrire le prédicat `reunis/3` qui, étant donné deux listes, crée une troisième liste dont les éléments sont des paires contenant un élément de chaque liste, dans l'ordre. Si l'une des listes contient plus d'éléments que l'autre, ceux-ci sont ignorés.

```
?- reunis([a,b,c],[1,2,3,4,5],R).
R = [[a,1],[b,2],[c,3]]
```

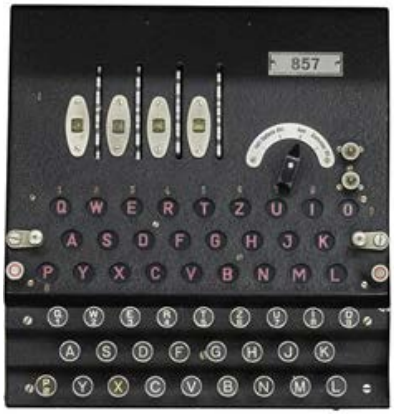
Exercice 4 : Palindrome

Ecrire le prédicat `palindrome/1` qui détermine si un mot est un palindrome.

```
?- palindrome(ressasser).
true
```

```
?- palindrome(ressasse).
false
```

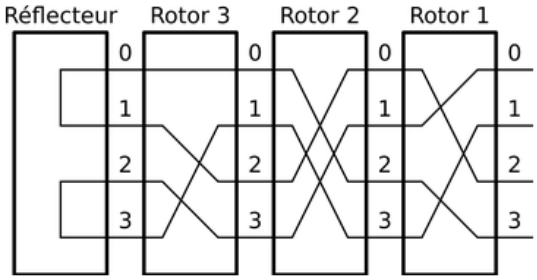
Exercice 5 : Enigma



Le but de cet exercice est de simuler la machine Enigma, utilisée par l'armée allemande pendant la seconde Guerre Mondiale pour chiffrer et déchiffrer leurs messages. Ce mécanisme de chiffrement réputé inviolable a été cassé par les Polonais puis en Angleterre par l'équipe d'Alan Turing, un des pères de l'informatique (voir le film « *Imitation Game* »).
Cette machine visait à transformer chaque lettre d'un message en une autre lettre. Chaque fois que l'utilisateur appuie sur une touche, une des lettres situées au-dessus s'allume (voir photo ci-contre). Pour déchiffrer, on réalise une opération similaire : on tape le message chiffré sur le clavier et les lettres du message en clair s'allument au fur et à mesure.

La machine utilise des rotors (visibles sur la photo) qui relient chacun électriquement une lettre en entrée avec une lettre en sortie.

Voici un exemple simple avec uniquement 4 chiffres au lieu des 26 lettres. Le chiffre à coder est à droite (par exemple, 0). Les sorties du rotor 1 sont reliées avec les entrées du rotor 2. Les sorties du rotor 2 sont reliées aux entrées du rotor 3. Les sorties du rotor 3 sont reliées à un réflecteur qui, lui aussi, relie différemment les chiffres et renvoie sur le rotor 2 puis le rotor 1. Sur l'exemple ci-contre, le 0 en entrée est ainsi relié avec le 1. Si on appuie sur le 0, c'est le 1 qui va s'allumer. Pour déchiffrer le message, il suffit d'appuyer sur le 1 et le 0 va s'allumer puisque la machine, dans cette configuration a une connexion électrique entre le 0 et le 1.



a) On suppose qu'on utilise deux rotors seulement et qu'ils sont fixes. On représente chaque rotor par une liste associant chaque lettre de départ avec une lettre d'arrivée. Voici la description de 2 rotors et d'un réflecteur :

```
rotor(1,[[a,e],[b,l],[c,w],[d,m],[e,h],[f,a],[g,s],[h,n],[i,o],[j,i],[k,x],[l,y],[m,b],[n,r]
,[o,z],[p,q],[q,d],[r,c],[s,v],[t,g],[u,t],[v,f],[w,k],[x,p],[y,u],[z,j]]).

rotor(2,[[a,d],[b,k],[c,s],[d,c],[e,l],[f,p],[g,b],[h,x],[i,w],[j,o],[k,t],[l,g],[m,u],[n,j]
,[o,f],[p,z],[q,e],[r,v],[s,r],[t,q],[u,m],[v,i],[w,a],[x,y],[y,n],[z,h]]).

reflecteur([[a,m],[b,q],[c,e],[d,t],[e,c],[f,z],[g,w],[h,u],[i,n],[j,l],[k,y],[l,j],[m,a],[n
,i],[o,v],[p,x],[q,b],[r,s],[s,r],[t,d],[u,h],[v,o],[w,g],[x,p],[y,k],[z,f]]).
```

Copier-coller ces lignes et écrire le prédicat **coder_message**/2 qui étant donné une liste de lettres, produit une liste de lettres chiffrées. Par exemple, la lettre **a** doit être codée par **h** (de a à e par le rotor 1, de e à l par le rotor 2, de l à j par le déflecteur, de j à n par le rotor 2 inversé et de n à h par le rotor 1 inversé). Vous pouvez utiliser le prédicat member/2 pour récupérer directement la bonne sous-liste dans la liste totale.

Utilisez votre programme pour décoder le message suivant attribué à Albert Einstein :
[s,h,e,x,v,y,v,l,u,y,d,k,k,v,s,v,e,v,s,d,x,s,w,h,t,u,h,e,h,z,y,v,g,q,d,t,g,z,v,q,h,l,u,d,k,j,v,g]

b) La machine était réputé inviolable parce que les rotors changent après chaque lettre en décalant d'un cran les entrées et les sorties. Ainsi, après le codage de la première lettre, le rotor 1 est décalé d'un cran pour devenir : [[a,l],[b,w],[c,m],[d,h],..., [y,j], [z,e]].

Ecrire le prédicat **decaler**/2 qui étant donné une liste de listes de 2 éléments, produit une liste dans laquelle les seconds éléments ont été décalés d'un cran vers la gauche.

Par exemple :

```
?- decaler([[a,e],[b,m],[c,z]],L).
L = [[a, m], [b, z], [c, e]].
```

c) A partir du prédicat coder_message/2, écrire le prédicat **coder_message1**/2 qui prend en compte ce décalage d'un cran du rotor 1 après chaque chiffrement d'une lettre. Il vous faudra probablement passer en paramètre les listes des rotors.

Pour tester décodez ce message : [j,k,w,z,h,c,n,c,z,r,t,k,d,j,a,c,d,k,f,g,i]

d) Lorsque le rotor 1 a fait un tour complet, il décale d'un cran le rotor 2, comme dans un compteur kilométrique de voiture. A partir du prédicat coder_message1/2, écrire le prédicat **coder_message2**/2 qui prend en compte cette nouvelle particularité. Vous devriez alors pouvoir décoder les messages suivants attribués respectivement à Pierre Dac et à Albert Einstein :

```
[x,j,g,l,y,x,y,t,f,u,s,j,j,a,r,f,w,m,u,p,f,j,z,g,a,p,a,m,d,y,a,m,s,d,s,d,v,q,j,i,o,l,r,y,m
,l,q,n,i,b,n,r,b,y,u,d,k,o,a,v,a,l,i,w,f,n,k,w,r,c,f,j,m,j,b,p,w,t,y,i,h,x,k,e,l,r,q,l,z,r
,g,g,t,z,p,s,g,t,r,y,v,z,x,j,j,p,r,f,x,m,o,g,d,f,g,o,t,z,f,d,c,q,r,o,t,i,i,r,p,n,a]

[s,i,c,j,t,p,c,j,c,j,t,d,k,b,j,b,y,c,z,y,q,j,z,s,i,a,s,r,t,b,o,j,s,d,v,d,h,z,r,h,r,q,e,h,b,p
,o,m,x,k,m,a,b,q,o,g,y,o,q,e,a,i,x,g,m,n,d,z,v,y,q,p,z,q,p,w,d,e,x,w,o,c,p,g,g,w,o,c,n,o,y,q
,p,k,x,t,u,f,t,g,u,z,y,j,x,w,z,u,m,s,j,z,w,o,g,r,t,j,x,g,w,z,x,w,w,v,i,a,k,w,q,l,a,c,o,s,b,u
,d,v,b,q,f,f,k,z,z,v,l,u,q,r,h,j,k,j,d,s,u,s,r,i,q,x,i,r,c,s,z,d,c,o,d,u,s,l,z,h,m,r,q,v,w,v
,p,t,w]
```