

# Examen de programmation logique

5 mai 2020

La clarté et la précision de la rédaction seront prises en compte dans l'évaluation.  
Le barème est indicatif.

Aucune question ne pourra être posée durant l'examen.  
En cas de doute concernant le sujet,  
vous poursuivrez votre réponse en expliquant vos hypothèses.

Vous déposerez votre copie au format électronique (.pdf ou .pl) dans l'onglet travaux d'Arel,  
au maximum 10 minutes après la fin de l'épreuve.

Si vous devez rendre votre travail par mail, merci d'utiliser comme sujet :  
**[examen prolog] NOM Prénom groupe**

Durée : 2h

## Traduction /2 pt

Traduire en prolog les énoncés suivants :

1. Alice est confinée.
2. Bob est le mari d'Alice.
3. Des époux sont confinés ensemble.
4. Avec qui est confinée Alice ?

## Persistence multiplicative /4 pt

La persistence multiplicative d'un nombre est le nombre d'étapes nécessaires pour obtenir un nombre à un chiffre en multipliant successivement les chiffres du nombre.

Par exemple,  $679 \rightarrow 6 * 7 * 9 = 378 \rightarrow 3 * 7 * 8 = 168 \rightarrow 1 * 6 * 8 = 48 \rightarrow 4 * 8 = 32 \rightarrow 3 * 2 = 6$   
La persistence multiplicative de 679 est donc de 5 (5 étapes pour atteindre le chiffre 6).

Écrire un prédicat `entier_persistence(E,N)` qui est vrai si la persistence multiplicative du nombre entier E vaut N.

```
?- entier_persistence(679,N).  
N = 5.
```

## Listes /4 pt

Écrire le prédicat `liste_sommeMax(L,N)` qui est vrai si la somme maximale de deux éléments non consécutifs de la liste d'entier L vaut N.

```
?- liste_sommeMax([26,17,3,14,5,16,8],N).  
N = 42.
```

## Contraintes /4pt

Soit la concaténation des liste des chiffres utilisée pour écrire le carré et le cube d'un entier donné.

Écrire le prédicat `entierCarreCube(N)` qui est vrai si la liste associée à `N` est composée d'exactly une fois chaque chiffre de 0 à 9.

```
?- entierCarreCube(N).
```

```
N = 69.
```

## Problème /6pt

On rappelle que le nombre de combinaisons de  $k$  objets parmi  $n$  vaut

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Écrire un prédicat `m_borne_nbElements(M,Borne,NbElements)` qui est vrai si `NbElements` est le nombre de combinaisons  $\binom{n}{k}$  avec  $0 \leq n \leq M$  et  $0 \leq k \leq n$  qui sont au moins égales à `Borne`.

Que donne `?- m_borne_nbElements(1000,1000000,NbElements)?`

