

Examen de programmation logique

1er juin 2016

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

Le barème est indicatif.

Aucun document autorisé.

Machines (ordinateurs, téléphones, montres connectées et calculatrices) interdites.

Aucune question ne pourra être posée durant l'examen. En cas de doute concernant le sujet, vous expliquerez vos hypothèses.

Durée : 2h

1 Mise en jambes /4 pt

Écrire les 2 prédicats suivants :

1. **dernier** qui retourne vrai si la valeur en sortie (deuxième paramètre) est le dernier élément d'une liste en entrée (premier paramètre).

```
?- dernier([1,3,6,3,2],E).
```

```
E = 2
```

2. **sommeliste** qui retourne vrai si la valeur en sortie (deuxième paramètre) est la somme des éléments d'une liste en entrée (premier paramètre).

```
?- sommeliste([1,3,6,3,2],S).
```

```
S = 15
```

2 Somme et différence /4 pt

Écrire un prédicat **diffsomme** qui retourne vrai si la valeur en sortie (second paramètre) contient la somme des éléments de rang pair moins celle des éléments de rang impair d'une liste en entrée (premier paramètre). La tête de la liste est au rang 1.

```
?- diffsomme([1,3,6,3,2],L).
```

```
L = -3
```

3 Compression /4 pt

Écrire un prédicat `compress` qui retourne vrai si la liste en sortie (second paramètre) contient les éléments de la liste en entrée (premier paramètre), en remplaçant les éléments identiques successifs par un seul exemplaire de cet élément.

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

4 Animaux mutants /4 pt

Écrire un prédicat `mutant` qui retourne vrai si la liste en sortie (troisième paramètre) représente un animal mutant, qui peut être obtenu à partir des deux listes en entrée (premier et deuxième paramètre) qui représentent deux animaux. Nous disposons d'un prédicat `animal` pour vérifier qu'une liste est un animal. Un animal mutant peut être obtenu à partir de deux animaux si un suffixe non vide du premier est préfixe du second. Par exemple, cheval et albatros peuvent engendrer le mutant chevalbatros.

```
?- mutant([c,h,e,v,a,l],[a,l,b,a,t,r,o,s],Mutant).  
Mutant = [c,h,e,v,a,l,b,a,t,r,o,s]
```

5 Coloration /4 pt

Écrire un prédicat `coloration` qui retourne vrai si la liste en sortie (second paramètre) représente la coloration d'un graphe de N sommets numérotés de 1 à N (premier paramètre) avec le minimum de couleurs. Deux sommets adjacents ne peuvent pas avoir la même couleur. On considèrera que le graphe est modélisé par les prédicats `arete(X,Y)`, indiquant qu'il y a une arête entre les sommets X et Y . Les couleurs seront codées par des entiers consécutifs commençant à 1.

```
?- listing(arete),coloration(4,C).  
arete(1, 2).  
arete(2, 3).  
arete(3, 4).  
arete(4, 1).
```

```
C = [1, 2, 1, 2]
```

Bonus

Trouver un animal mutant de degré 4, c'est-à-dire issu du croisement de 4 animaux selon la règle de l'exercice 4. Par exemple vachevalbatros est un animal mutant de degré 3.

