

Examen de programmation logique

16 mai 2019

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 poursuivrez votre réponse en expliquant vos hypothèses.

Durée : 2h

Questions de cours /1 pt

1. Quel symbole permet d'appliquer la coupure (cut) en `prolog` ?
2. Quelle propriété possède l'opérateur `#=` que ne possède pas l'opérateur `is` ?

Traduction /2 pt

Traduire en `prolog` les énoncés suivants :

1. Alice communique avec Bob.
2. Bob est un pirate.
3. Un pirate espionne les personnes avec qui il communique.
4. Qui espionne qui ?

Arithmétique /3 pt

Écrire un prédicat `entier_entier_ppcm(N,M,R)` qui retourne vrai si R est le plus petit multiple commun de N et M.

```
?- entier_entier_ppcm(6,8,R).  
R = 24.
```

Listes /4 pt

1. Écrire un prédicat `verifListe_borneInf_borneSup(L,Binf,Bsup)` qui est vrai si les éléments de la liste L sont tous compris (au sens large) entre les entiers BInf et Bsup.

```
?- verifListe_borneInf_borneSup([4,3,6,2],1,6).  
true  
?- verifListe_borneInf_borneSup([4,3,6,2],1,5).  
false
```
2. Écrire un prédicat `liste_listeASupprimer_listeReduite(L1,L2,L3)` qui est vrai si la liste L3 contient les éléments de la liste L1 qui ne sont pas dans la liste L2.

```
?- liste_listeASupprimer_listeReduite([a,b,c,d,e],[a,d],L).  
L = [b,c,e]
```

Contraintes /3pt

1. Écrire un prédicat permettant de résoudre le problème suivant : trouver deux carrés de 2 chiffres tels que le nombre de 4 chiffres formé en les accolant soit aussi un carré (par exemple 16,81 et 1681). Vous pourrez bien évidemment écrire des prédicats intermédiaires.

Conjecture de Sheldon /7pt

1. Écrire un predicat `niemePremier(N,I)` qui retourne vrai si N est le I^{eme} nombre premier.
2. Écrire un prédicat `proprietesProduit(N)` qui retourne vrai si N vérifie les propriétés suivantes :
 - (a) N est le I^{eme} nombre premier
 - (b) I est le produit des chiffres de N (en base 10).

Par exemple 73 vérifie la propriété car il s'agit du 21^{eme} nombre premier et $21 = 3 * 7$

3. Écrire un prédicat `entierMiroir(N,R)` qui retourne vrai si R est le nombre obtenu à partir de N en inversant l'ordre des chiffres, par exemple 73 et 37.
4. Écrire un prédicat `proprietesMiroir(N)` qui retourne vrai si N vérifie les propriétés suivantes :
 - (a) N est le I^{eme} nombre premier
 - (b) Si R est le miroir de N alors il s'agit du J^{eme} nombre premier, avec J miroir de I .

Par exemple 73 vérifie la propriété car il s'agit du 21^{eme} nombre premier et que son miroir, 37 est le 12^{eme} nombre premier.

