



Université d'Ottawa • University of Ottawa

## Faculté de Génie – Département d'informatique

### CSI2520 : PARADIGMES DE PROGRAMMATION

#### Hiver 2014 – Solution Laboratoire 1

Tester le programme suivant (il pourrait être enrichi avec des faits supplémentaires):

Soit la base de faits Prolog suivante :

```
professeur(turing, csi2520).  
professeur(codd, csi4533).  
professeur(backus, csi2511).  
professeur(ritchie, csi2772).  
professeur(minsky, csi2510).  
professeur(codd, csi2530).
```

```
etudiant(fred, csi2520).  
etudiant(paul, csi4533).  
etudiant(jean, csi2510).  
etudiant(jean, csi2772).  
etudiant(henri, csi2510).  
etudiant(henri, csi2530).
```

```
annee(fred, 1).  
annee(paul, 2).  
annee(jean, 2).  
annee(henri, 4).
```

1. Quel sera le résultat de la requête suivante :

```
?- professeur(codd, Cours), etudiant(UnEtudiant, Cours).
```

2. Donner toutes les solutions qui seront obtenues, dans l'ordre où elles seront trouvées.

```
Cours = csi4533,  
UnEtudiant = paul ;  
Cours = csi2530,  
UnEtudiant = henri.
```

3. Ecrire la requête permettant de trouver les étudiants de minsky.

```
?- professeur(minsky,C),etudiant(UnEtudiant,C).  
C = csi2510,  
UnEtudiant = jean ;  
C = csi2510,  
UnEtudiant = henri.
```

4. Ecrire la requête permettant de trouver les cours suivis par des étudiants de 4ème année

```
UnEtudiant = henri,  
Cours = csi2510 ;  
UnEtudiant = henri,  
Cours = csi2530.
```

5. Ecrire la requête permettant de trouver les étudiants qui ont le même professeur.

```
?- etudiant(UnEtudiant,Cours1), etudiant(UnEtudiant2,Cours2),  
UnEtudiant\=UnEtudiant2, professeur(P,Cours1),  
professeur(P,Cours2).  
UnEtudiant = paul,  
Cours1 = csi4533,  
UnEtudiant2 = henri,  
Cours2 = csi2530,  
P = codd ;  
UnEtudiant = jean,  
Cours1 = Cours2, Cours2 = csi2510,  
UnEtudiant2 = henri,  
P = minsky ;  
UnEtudiant = henri,  
Cours1 = Cours2, Cours2 = csi2510,  
UnEtudiant2 = jean,  
P = minsky ;  
UnEtudiant = henri,  
Cours1 = csi2530,  
UnEtudiant2 = paul,  
Cours2 = csi4533,  
P = codd ;  
false.
```

6. Créer un prédicat 'note' donnant le résultat obtenu par un étudiant dans un cours. Les notes possibles sont a,b,c,d,e, f.

```
note(fred,csi2520,a).  
note(paul, csi4533,b).  
note(jean, csi2510,c).  
note(jean, csi2772,d).  
note(henri, csi2510,e).  
note(henri, csi2530,f).
```

7. Ecrire un prédicat 'reussit(etudiant,cours)' vérifiant si un étudiant a passé un cours, ce prédicat doit vérifier si :

- a. l'étudiant est inscrit à ce cours
- b. et s'il a obtenu une note autre que e ou f

```
reussit(Etudiant,Cours):-  
etudiant(Etudiant,Cours),note(Etudiant,Cours,N),N\=e,N\=f.
```