



Analyse d'un programme PROLOG

Un programme Prolog commence par les lignes suivantes:

```
member(X, [X|_]).
```

```
member(X, [_|_]) :- member(X, _).
```

*Question 1: De quel type sont les arguments du prédicat **member**?*

*Question 2: Que fait le prédicat **member**? Indiquez la propriété qui est toujours vérifiée entre ses arguments.*

Le programme continue avec les lignes suivantes:

```
reverse(L, R) :- reverse(L, [], R).
```

```
reverse([], R, R).
```

```
reverse([_|_], S, R) :- reverse(_, [_|_], S), reverse(S, R).
```

*Question 3: De quels types sont les arguments du prédicat **reverse**? Que remarquez-vous?*

*Question 4: Que fait le prédicat **reverse**? Indiquez la propriété qui est toujours vérifiée entre ses arguments.*

Le programme continue avec les lignes suivantes:

```
path(Z, Z, L, R) :- reverse(L, R).
```

```
path(X, Y, L, R) :- move(X, Z), not(member(Z, L)), path(Z, Y, [_|_], R).
```

```
move(1, 6).
```

```
move(7, 2).
```

```
move(6, 7).
```

*Question 5: De quels types sont les arguments du prédicat **path**? Tenez compte des réponses aux questions précédentes pour préciser.*

*Question 6: Quelle est la réponse à la requête **path(1, 2, [_], R)** ?*

On ajoute au programme les lignes suivantes:

```
move(1, 8).
```

```
move(2, 7). move(2, 9).
```

```
move(3, 4). move(3, 8).
```

```
move(4, 3). move(4, 9).
```

```
move(6, 1).
```

```
move(7, 6).
```

```
move(8, 3). move(8, 1).
```

```
move(9, 4). move(9, 2).
```

*Question 7: Quelle est maintenant la réponse à la requête **path(1, 2, [_], R)** ?*

*Question 8: Plus généralement, que provoque la requête **path(P, Q, [_], R)** ? Il n'est pas demandé de calculer la réponse, mais de décrire en quelques phrases ce qu'elle exprime.*

Durée: 1 heure 30 minutes

Les Documents et calculettes sont autorisés, mais ne doivent en aucun cas circuler entre les étudiants