



Durée : 60 minutes. 1 document personnel manuscrit A4 autorisé. Sujet sur 1 recto simple.

I. Exécutions et arbre d'exécution (barème indicatif : 7 points)

La spécification du programme suivant a été perdue.

```
calcul([ ], R, [ ]).  
calcul(L, [ ], L).  
calcul([E|L], [E|R], T) :- calcul(L,R,S), calcul(S,R,T).  
calcul([E|L], R, [E|S]) :- calcul(L,R,S).
```

Q1. Indiquez le/s résultat/s des requêtes suivantes (il n'est pas demandé de justifier ou il n'est pas demandé d'arbre d'exécution) :

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

Q2. Dessiner le début de l'arbre d'exécution de la requête suivante (dessiner au plus 10 nœuds ou étapes) :
?- calcul([1,2],[1,2],R).

II. Complétude, correction, terminaison (barème indicatif : 6 points)

Le programme suivant prétend faire la concaténation de deux listes (en premier paramètre le résultat de la concaténation ; les deux paramètres, ensuite, donnent les deux listes à concaténer). Ex. ?- conc(L,[1,2,3],[4,5]) doit donner L=[1,2,3,4,5].

```
conc([ ],[ ],[ ]).  
conc([E|L],[E|L],[ ]).  
conc([E|R],[ ],[E|T]) :- conc(R,[ ],T).
```

Pouvez-vous, en quelques mots, dire si/pourquoi ce programme est :

- Q1.** Complet
- Q2.** Correct
- Q3.** Termine

Et s'il vous apparaît que ce programme échoue à l'un de ces critères, pouvez-vous donner un contre-exemple, et corriger le programme. (=> **Q4.** [si le programme est faux])

III. Tri en vitesse (barème indicatif : 7 points)

Un ingénieur pense avoir trouvé un nouvel algorithme de tri croissant plus rapide que le tri rapide ! L'idée est la suivante : dans un premier temps il veut prendre les entiers de la liste à trier deux par deux et placer le plus grand des deux dans une sous liste des « grands » entiers et le plus petit des deux dans une sous liste des « petits » entiers. Il pense ainsi obtenir deux sous-listes : la liste des « grands » entiers et la liste des « petits » entiers. Bien sur, la liste des « grands » entiers peut comporter des petits entiers plus petits que les plus grands entiers de la liste des « petits » entiers, mais globalement, pense-t-il, le tri a commencé. Ensuite, il pense recommencer récursivement sa méthode de tri avec chaque sous liste et à la fin remettre tout ensemble.

Q1. Spécifiez et réalisez un prédicat qui réalise la première étape de l'algorithme proposé (séparation d'une liste en 2 sous-listes : les « grands » entiers et les « petits » entiers ; pour prendre les entiers deux par deux, vous pouvez sélectionner les entiers deux par deux à la suite : le premier et le second, ou aux extrémités de la liste : le premier et le dernier, ou comme vous voulez, faites simple et efficace, il faut aller vite!)

Q2. La fin de l'algorithme n'est pas claire, mais comme vous êtes un grand spécialiste des tris récursifs, vous saurez trouver comment faire ... Spécifiez et réalisez un prédicat qui réalise le tri croissant recherché en utilisant judicieusement la première étape de séparation.

Q3. Votre avis, est-ce que l'algorithme obtenu est plus rapide que le tri rapide ?