

Durée : 80 minutes. 1 document personnel manuscrit A4 autorisé. Sujet sur 1 recto simple.

I. Exécutions et arbre d'exécution (barème indicatif : 6 points)

La spécification du programme suivant a été perdue.

```
modifie([], L, L).
modifie([E|L], Q, [E|R]) :- modifie(L, [E|Q], R).
```

Q1. Indiquez le/s résultat/s des requêtes suivantes (ne pas justifier) :

```
?- modifie([1,2,3], [], R).
?- modifie([1,2], [a,b], R).
?- modifie(L, [], [A,B,C,D]).
```

Q2. Choisissez l'une des requêtes précédentes et dessinez son arbre d'exécution.

II. Tri-Répartition de résultats (barème indicatif : 8 points)

L'objectif de cet exercice est de répartir et éventuellement de trier des résultats numériques en 3 classes :

- la classe des résultats négatifs,
- la classe des résultats normaux (à trier) et
- la classe des résultats excessifs dépassant la limite supérieure.

Les résultats en question sont des nombres et leurs valeurs normales sont comprises entre 0 et 100. En deçà de 0, les résultats sont (dits) négatifs et doivent être conservés dans l'ordre où ils sont apparus, ils doivent être placés en début ; entre 0 et 100 les résultats sont dits normaux et il faut les trier par ordre croissant, ils doivent apparaître ensuite dans la liste des résultats ; au delà de 100, les résultats sont dits excessifs et doivent être conservés dans l'ordre où ils sont apparus, ils doivent apparaître en fin.

Exemple : la liste en entrée : 12 -34 5 678 -9 0 doit donner la liste en sortie -34 -9 0 5 12 678 avec la classe des résultats négatifs en premier (-34 -9), puis la classe triée des résultats normaux (0 5 12) et enfin la classe des résultats excessifs (678).

Pour déterminer la classe d'un résultat, vous pouvez utiliser les 3 prédicats suivants :

```
negatif(X) :- { X < 0 }.
normal(X) :- {X >= 0, X <= 100}.
excessif(X) :- { X > 100}.
```

Q1. Dans un premier temps, résoudre le problème *sans trier* les résultats normaux. **Attention** : commencer par spécifier votre programme.

Q2. Résoudre le problème complet avec tri des résultats normaux. Commencer par spécifier votre programme.

III. Dérivation à analyser et corriger (barème indicatif : 6 points)

La dérivée d'une liste est la liste des différences entre les éléments successifs de la liste.

Exemple : la liste en entrée 6 5 5 3 6 a pour liste dérivée : -1 0 -2 3 (c'est à dire 5-6, 5-5, 3-5, 6-3)

Spécification du programme : `deriver(L,D)` vrai ssi L a pour dérivée D.

Q1 A et B et C. Analyser les programmes A, B, C suivants et donner leurs propriétés (correction, complétion, terminaison). Pour chaque propriété, donner une ligne d'explication.

Q2 A ou B ou C. Choisir l'un des programmes incorrects et corriger les erreurs qu'il comporte. Commentez vos corrections et ajouts.

Prog A :

```
deriver([ ], [ ]).
deriver([E], [E]).
deriver([E,F|L],R) :- {S=F-E}, deriver(L,[S])
```

Prog B :

```
deriver([ ], R).
deriver([E,F,L],[S|R]) :- {S=F-E}, deriver(L,R).
```

Prog C :

```
deriver([E], [E]).
deriver([E,F|L], R) :- {F=S-E}, deriver(L,[ ]).
```