

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.

```
stratégie(0, L, papier).
stratégie(N, [E], E).
stratégie(1, [E|L], R) :- stratégie(0, L, R).
stratégie(N, [E|L], ciseaux) :- stratégie(1, L, R).
```

**Q1.** Indiquez le/s résultat/s des requêtes suivantes (ne pas justifier, ni donner l'arbre d'exécution) :

```
?- stratégie(0, [pierre], R).
?- stratégie(1, [pierre], R).
?- stratégie(2, [pierre, pierre], R).
```

**Q2.** Dessiner le début de l'arbre d'exécution de la requête suivante (dessiner au plus 10 nœuds ou étapes) :

```
?- stratégie(N, L, R).
```

### II. Séparation déclarative (barème indicatif : 7 points)

L'objectif de cet exercice est de séparer une liste en 2 sous-listes : la sous-liste des premiers éléments et la sous-liste des derniers éléments. Si la liste de départ est de longueur paire, les deux sous-listes extraites seront de même taille. Si la liste de départ est de longueur impaire, l'une des sous-listes sera de longueur un peu plus grande (par exemple, la première sous-liste). Pour effectuer cette séparation, deux prédicats déclaratifs seront utilisés : le prédicat de concaténation (vue en cours, inverse de la séparation), et un prédicat pour établir que les sous-listes sont approximativement de même longueur.

**Q1.** Rappelez la définition du prédicat de concaténation vue en cours. Spécification, exemple et code.

**Q2.** Spécifiez et réalisez un prédicat qui lie deux listes pour qu'elles soient approximativement de même longueur (éventuellement, la première des deux listes peut avoir un élément de plus que la seconde). Spécification, exemple et code. Consigne : ne pas calculer la taille des listes et sous-listes, il y a une solution beaucoup plus simple et élégante qui consiste à parcourir les deux listes en même temps.

**Q3.** En utilisant les deux prédicats précédents, résoudre le problème de séparation d'une liste en 2 sous-listes de tailles égales ou presque (pour le cas où la liste de départ est de longueur impaire) . Spécification, exemple et code. Indiquez, avec le maximum de détails pourquoi votre code termine dans le cas d'une requête avec une liste initiale de taille connue, paire ou impaire.

### III. Finitions (barème indicatif : 7 points)

La photocopieuse a fourni un paquet de feuilles correspondant à l'impression en plusieurs exemplaires d'un document de plusieurs pages.

Malheureusement, la sélection des menus de l'imprimante n'a pas été faite correctement, et le paquet n'est pas ordonné comme il faut, il fallait « trié » et c'est « groupé » qui a été utilisé.

En haut du tas, il y a toutes les premières pages,

puis toutes les pages secondes, puis etc. (voir illustration, crédit Canon). Il faudrait redistribuer le paquet pour reconstituer les différents documents : en haut une première version du document, puis en dessous une seconde version du document, etc.

Modéliser la situation et proposer un algorithme pour remettre le tas « groupé » en ordre « trié ». Il semble naturel de séparer le problème global en plusieurs étapes ou sous-problèmes.

**Q1.** Spécifier et donner des exemples d'utilisation pour les prédicats que vous allez utiliser

**Q2.** Rédiger le programme principal.

**Q3.** Rédiger au moins l'un des sous-programmes réalisant une étape intermédiaire non triviale ou résolvant l'un des sous-problèmes.

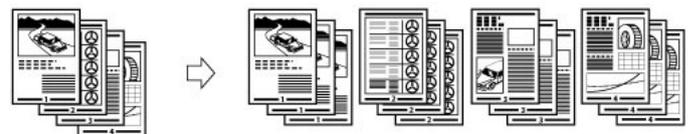


Figure 1: Photocopies groupées



Figure 2: Photocopies triées