G. HUNAULT Angers, 1999

Mini-Projet en Algorithmique et Programmation : Détection de parenthéseurs (Licence MASS)

1. Position du problème et Algorithmes

Deux éléments d'une liste, notés u et v, sont dits parenthéseurs pour la liste s'ils vérifient les conditions suivantes :

- il y a autant de u que de v dans la liste,
- le i-ème u dans la liste est strictement avant le i-ème v dans la liste

On cherche à détecter les parenthéseurs communs possibles pour plusieurs listes données. La méthode proposée est la suivante : d'abord se restreindre aux couples (u, v) d'éléments ayant le même nombre d'occurences pour chacune des listes. Ensuite ne retenir parmi ces couples que ceux dont les positions des u sont strictement avant celles des v pour chaque liste.

1.a) Pour chacune des 3 séries de 2 listes suivantes, indiquer les éventuels couples de parenthéseurs possibles.

Série 1	`a b c a` `a b c b`
Série 2	`12 57 57 12 21 100 21` `57 12 21 12 21 100`
Série 3	`a POM a d c c` `a POM d c POM a d c`

1.b) Soit Nbl le nombre de listes utilisées, mises à raison d'une liste par ligne dans un tableau $Tc[i]$ pour i de 1 à Nbl. Ecrire un algorithme qui détermine le nombre $NbSymb$ d'éléments différents utilisés dans l'ensemble du tableau Tc.

1.c) On note $NbSymbPl[i, j]$ le nombre de de fois où on trouve l'élément numéro j dans la ligne numéro i. Ecrire l'algorithme qui remplit $NbSymbPl[i, j]$ à partir de Tc et $NbSymb$.

1.d) Ecrire l'algorithme qui trouve tous les couples d'éléments qui ont le même nombre d'occurences pour chaque ligne de Tc. On notre $NbCpl$ le nombre de couples etCpl le tableau des couples parenthéseurs possibles.

1.e) Ecrire l'algorithme qui ne retient du tableau Cpl que les couples pour lesquels chaque occurence du premier élément du couple est strictement avant l'occurence correspondante du deuxième élément du couple.

2. Travail demandé

Vous devez tout d'abord écrire un algorithme principal et détailler la liaison entre cet algorithme et les sous-algorithmes. Vous détaillerez le choix du passage de paramètre(s) en utilisant la syntaxe vue en cours (module, paramètres, locales, globales...). Ensuite, vous viendrez traduire ces fonctions en Apl et en Rexx. Pour Apl, les listes seront dans une matrice de nombres exclusivement. Les listes seront les vecteurs lignes de la matrice, éventuellement complétées par des 0 (pour les listes courtes). Pour Rexx, les listes seront les lignes d'un fichier texte dont le nom sera passé en paramètre. Les éléments seront alors les mots de chaque ligne.

Ce miniprojet est à rédiger seul(e). Vous avez certainement à effectuer une recherche bibliographique simple (indiquez vos sources). Vous m'enverrez en attaché par mail un fichier .zip qui contient tout votre travail, à savoir : une zone de travail Apl (fichier .ws = "$Workspace$"), un fichier texte du programme Rexx, ainsi qu'un document au sens de Word pour $Windows$, au format RTF ou LateX qui contiendra les algorithmes, la documentation et vos commentaires.

Ces fichiers auront pour nom vos initiales : Jomes Band fournira donc JB.WS, JB.REX et JB.DOC le tout dans JB.ZIP ; par contre Nona Higane rendra NH.WS, NH.REX et NH.DOC. Vous ne mettrez aucun sous-répertoire.

Ce travail de recherche, programmation et rédaction est à rendre pour le mardi 07 décembre, 9h (aucun retard ne sera toléré).

Il vaut mieux rendre un travail incomplet que de ne rien rendre. La qualité des algorithmes, en particulier leur simplicité et leur lisibilité sera appréciée.

Indiquez clairement vos choix : si un terme vous parait ambigu, précisez sa définition pour vous. Justifiez vos modes de programmation ; récursivité ou linéarité, appel d'opérateurs, de fonctions prédéfinies ou écriture de boucles explicites. Nommez distinctement vos variables (en particulier, vous utiliserez systématiquement au moins 3 lettres par identificateur). N'hésitez pas à commenter sans dépasser toutefois un commentaire toutes les 8 lignes en moyenne (sauf pour l'en-tête).

Vous n'êtes pas obligé(e) de tout tester dans tous les sens. Effectuez donc des tests élémentaires. L'affichage des résultats à l'écran devra être très propre, très soigné. Ainsi en cas d'affichage numérique multi-lignes, on mettra bien les unités sous les unités, les dizaines sous les dizaines etc. Soignez aussi la rédaction, l'orthographe...