PCSI - exercices d'informatique

Séquences, listes, ensembles

Exercice 1

On donne une liste $a = [a_1, a_2, a_3]$ et on rappelle que son terme d'indice i se note a[i] en *Maple*©.

Former la séquence suivante :

$$a_1, a_2, a_3, 2a_1, 2a_2, 2a_3, 3a_1, 3a_2, 3a_3, 4a_1, 4a_2, 4a_3.$$

[On utilisera deux instructions seq imbriquées.]

Exercice 2

La procédure convert transforme les listes en ensembles et vice-versa selon la syntaxe suivante :

> convert([a, b, c],set); retourne {a, b, c}.
> convert({1, 2, 3},list); retourne [1, 2, 3].

Que se passe-t-il lorsque l'on convertit

(a) Un ensemble en liste, puis cette liste à nouveau en ensemble ? Donner un exemple.

(b) Une liste en ensemble, puis cet ensemble à nouveau en liste ? Idem.

Exercice 3

Ecrire une fonction retourne prenant comme argument une liste L et renvoyant la liste formée des mêmes éléments en ordre inverse.

Exercice 4

Ecrire une fonction recolle prenant pour arguments deux listes et effectuant la fusion ou concaténation de celles-ci.

Appliquée aux listes $[a, b, c, d]$ et $[e, f, g, h]$, cette fonction doit renvoyer la liste $[a, b, c, d, e, f, g, h]$.

Exercice 5

Ecrire une fonction echange(L,i,j) prenant comme arguments une liste L et deux indices i et j, et retournant la liste obtenue en échangeant les termes de L correspondant aux indices i et j.

Exercice 6

Ecrire une fonction estPresent(L,a) prenant comme argument une liste L et un objet a et retournant *true* ou *false* selon que a figure ou non parmi les termes de la liste L.

Exercice 7

Ecrire une fonction premier(L,a) prenant comme argument une liste L et un objet a et retournant l'indice de la première occurrence de a dans L. Si L ne contient pas a, la fonction répondra *false*.

Modifier cette fonction pour qu'elle retourne l'indice de dernière occurrence de a dans L.

Exercice 8

Ecrire une fonction compte(L,a) prenant comme argument une liste L et un objet a et retournant le nombre d'occurrences de a dans L.

Exercice 9

Dans cet exercice, on suppose que les données contenues dans la liste L sont de type *numeric*.

Ecrire une fonction maxi(L) prenant comme argument une liste L et retournant le plus grand élément de L.

Modifier cette fonction pour qu'elle retourne l'*indice* du plus grand élément de L.

Modifier à nouveau cette dernière fonction pour que, dans le cas où le plus grand élément de L serait présent plusieurs fois dans L, elle retourne la séquence de tous les indices correspondants.

Exercice 10 *tri "bulle" (bubble sort)*

Le *tri à bulle* d'une liste (contenant des données *numeric*) utilise le principe suivant :

On compare deux à deux les éléments consécutifs de la liste. S'ils ne sont pas dans l'ordre, on les échange. On répète le points précédents jusqu'à ce que plus aucun échange n'ait lieu. À la fin des opérations, les éléments les plus "légérs" (les plus petits) ont "remonté" (comme des bulles), et la liste est triée.

Ecrire une procédure triBulle() (sans argument) effectuant cet algorithme sur une liste L déclarée comme variable globale (on pourra réutiliser ex. 5).

Exercice 11

On donne

- un entier naturel $n \in \mathbb{N}^*$;
- une liste S de longueur n : $S = [s_1, ..., s_n]$;
- une liste $P = [p_1, ..., p_n]$ contenant les n entiers distincts de 1 à n dans un ordre quelconque (càd, P est une *permutation* de $[\![1, n]\!]$).

Ecrire une fonction permute(S,P) formant la liste T telle que pour tout k entre 1 et n,

$$t_k = s_{p_k}.$$

Par exemple si $S = [a, b, c, d]$ et $P = [3, 2, 4, 1]$, alors $T = [c, b, d, a]$.