

A rendre en deux feuilles séparées :

Feuille A : Exercices 1 et 2

Feuille B : Exercices 3 et 4

Exercice 1

Procédure supprimant les caractères ESPACE placés au début d'une chaîne

Ecrire une procédure appelée `DelSpace` admettant comme argument `chaîne`, qui est une chaîne de caractères se terminant par le caractère NUL, et indiquée par rapport à 0. Cette procédure supprime uniquement les espaces qui sont éventuellement présents au **début de la chaîne**. A l'issue de la procédure, le premier caractère de la chaîne ne sera pas un espace, mais la chaîne pourra quand même contenir des caractères espace. On évitera l'usage de fonctions externes.

Exercice 2

Conversion d'un nombre base 10 en base N

On désire écrire une procédure qui convertit un nombre écrit en base 10 en une base quelconque N. La méthode utilisée est la méthode des divisions successives. Les restes des divisions successives forment le nombre converti en base N, en commençant par les unités. Ainsi pour le nombre 977 converti en base 5, nous aurons :

$$977 : 5 = 195, \text{ Reste } 2$$

$$195 : 5 = 39, \text{ Reste } 0$$

$$39 : 5 = 7, \text{ Reste } 4$$

$$7 : 5 = 1, \text{ Reste } 2$$

$$1 : 5 = 0, \text{ Reste } 1$$

Il n'y a plus de division possible. On constitue les restes successifs sont alors : 2, 0, 4, 2, 1. En inversant leur ordre, on reconstitue le nouveau nombre : 12402.

On demande d'écrire l'algorithme d'une procédure appelée sous la forme :

`Cvt10n(valeur, table, base)`

Les arguments sont :

- `valeur` : contient le nombre décimal (entier) à convertir
- `table` : tableau de 32 entiers maximum (indiqué par rapport à 0)
- `base` : entier donnant la nouvelle base

A l'issue de l'appel de cette procédure, `valeur` contient le nombre de digits de la conversion (le nombre de restes chargés dans `table`), `table` contient la succession des restes, dans l'ordre du calcul, c'est-à-dire en commençant par les unités, et `base` reste inchangé. On dispose de l'opération modulo (`%`), qui donne le reste de la division entière.

Le tableau `table` est indiqué par rapport 0.

Exercice 3

Procédure d'inversion de l'ordre des valeurs dans un tableau d'entiers.

On désire écrire une procédure qui admet comme arguments `table`, un tableau d'entiers, et `N` le nombre d'éléments de ce tableau. Cette procédure, appelée `Inverse(table, n)` inverse l'ordre des éléments de `table`. Le tableau `table` est indicé par rapport à 0.

Exercice 4

Fonction de conversion d'un nombre écrit en base `N` en nombre base 10.

Écrire l'algorithme d'une fonction appelée comme suit :

`nombre = CvtN10(table, n, base)`

`table` est un tableau de `n` entiers indicé par rapport à 0. Il contient les digits du nombre exprimé dans la base précisée par l'argument `base`. Les digits sont organisés en commençant par les poids forts et en finissant par les unités. Cette fonction retourne la valeur de tableau convertie en base 10.

Remarque : L'organisation de `table` est inverse par rapport à celle générée dans l'exercice 2