AG43
Examen médian
Mardi 6 Novembre 2012
Durée 1h30

Exercice 1

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` sans utiliser de tableau supplémentaire.

Exercice 2

Nombre d'occurrences des lettres de l'alphabet

Ecrire une procédure nommée `alpha_occurrences` qui compte le nombre d'occurrences de chaque lettre de l'alphabet (nombre de fois où on les rencontre) dans une chaîne de caractères qui sera passée en paramètre (`chaine`), de même que le tableau des occurrences (`table`), qui est un tableau de 26 entiers.
On supposera que les chaînes sont terminées par le caractère NUL.
Vous pourrez utiliser (ou non, car il existe plusieurs méthodes) une chaîne interne qui contiendra les 26 lettres de l'alphabet dans l'ordre. Pour information, le code ASCII de 'A' est 65.
Vous disposez de la procédure `strupr(chaine)` qui convertit une chaîne en majuscules et les caractères accentués en caractères non accentués (exemple : é, e, ê deviennent E).
Les tableaux et chaînes sont indicés par rapport à 0. Ne pas oublier les initialisations.

Exemple :
Si `chaine` contient le mot "test", le tableau des occurrences `table` contiendra `[0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,2,0,0,0,0,0]` après appel de la procédure.

Exercice 3

Réduction d'un Buffer

On considère un buffer d'octets dont la longueur ne peut excéder 255 octets. Le format du buffer est le suivant :

```
[longueur n][octet 1][octet 2]...[octet n]
```

`n` est inférieur à 256.
Ce buffer est utilisé pour manipuler des nombres entiers codés en base 100.
Il arrive souvent, qu'après une opération comme une soustraction, la valeur contenue dans le buffer soit de plusieurs ordres de grandeur inférieure à sa valeur initiale. Dans ce cas, les

premiers octets (hormis l'octet 0 qui donne la longueur utile du buffer) contiennent la valeur 0. On souhaite écrire une procédure appelée `reduction(buffer)` qui supprime les premiers octets nuls, tout en conservant l'intégrité du format utilisé.

Exercice 4

Fonction de gestion d'horloge

Ecrire une procédure qui sera appelée chaque minute (ne pas se soucier de la procédure d'appel), chargée de gérer une horloge interne dans un microcontrôleur. Cette procédure ne prend pas d'argument et sera appelée `horloge()`.
L'horloge interne est en fait un tableau de 5 octets appelé `clock[]`, indicé par rapport à 0. Le tableau, variable globale, est organisé comme suit :
`[AA, MM, JJ, HH, mm]`
Les minutes sont comptées de 0 à 59, les heures de 0 à 23, les jours de 1 à n (n étant fonction du mois en cours), les mois de 1 à 12 et les années sont représentées par les 2 derniers chiffres du millénaire en cours. A chaque appel, la fonction incrémente l'octet des minutes, et gère les autres octets en conséquence (il se peut que l'on change d'heure, de jour, etc.). La procédure doit être valide pour tout le millénaire actuel.
On dispose de :

- La fonction `bis(annee)`, qui admet comme argument un entier qui est une année sous forme julienne (ex : 12 doit être passé sous la forme 2012) et qui retourne un entier, 1 pour une année bissextile, 0 pour une année normale.
- TM[] qui est un tableau d'octets, déjà initialisé, indicé par rapport à 0, contenant la durée de chaque mois pour une année normale.
  TM={31,28,31,30,31,30,31,31,30,31,30,31}