



Concours Toutes Options  
Epreuve d'Informatique

Date : Mardi 01 Juin 2010      Heure : 15 H      Durée : 2 H      Nbre pages : 4

Barème : EXERCICE 1 : 4 points (1.5, 2.5)  
EXERCICE 2 : 5 points (1, 1, 0.5, 1, 1.5)  
PROBLEME : 11 points (1, 2, 1, 2, 1, 2, 2)

DOCUMENTS NON AUTORISES  
L'USAGE DES CALCULATRICES EST INTERDIT

EXERCICE 1

1) On considère les deux suites  $u$  et  $v$  définies par :

$$\begin{cases} u_0 = 1 \\ u_k = \sqrt{2} \cos(kt) \end{cases} \quad k \in \mathbb{N}^* \quad \text{et} \quad \begin{cases} v_0 = 1 \\ v_k = \cos^k(t) \end{cases} \quad k \in \mathbb{N}^*$$

Donner les commandes MAPLE permettant de :

- 1.1) définir les deux suites  $u$  et  $v$  ;
- 1.2) calculer  $A$ , la valeur de  $u_1$  et  $B$  la valeur de  $v_1$  ;
- 1.3) évaluer  $A$  et  $B$  pour  $t = \frac{\pi}{3}$  ;
- 1.4) calculer puis simplifier les expressions suivantes :

$$\sum_{k=1}^n u_k v_k \quad \text{et} \quad \sum_{k=1}^n u_k \sum_{k=1}^n v_k$$

- 1.5) calculer les limites des deux suites  $u$  et  $v$  quand  $k \rightarrow +\infty$  ;

2) Soient  $(E1)$  et  $(E2)$  les équations différentielles définies par :

$$(E1): y'(t) + y(t) = u_1$$

$$(E2): 2y''(t) + y'(t) + \sin(t) = v_1$$

Donner les commandes MAPLE permettant de :

- 2.1) définir les deux équations différentielles ( $E1$ ) et ( $E2$ ) ;
- 2.2) calculer  $S1$ , la solution de la résolution formelle de l'équation différentielle ( $E1$ ) avec la condition initiale suivante :  $y(0) = 1$  ;
- 2.3) calculer  $S2$ , la solution de la résolution numérique de l'équation différentielle ( $E2$ ) avec les conditions initiales suivantes :  $y'(0) = 1$  et  $y(0) = 1$  ;
- 2.4) convertir  $S1$  en une fonction  $y$  ;
- 2.5) représenter graphiquement  $y$  pour  $t \in [-5..5]$  ;
- 2.6) représenter graphiquement  $S2$  pour  $t \in [-5..5]$ .

## EXERCICE 2

On relève dans certains jours de l'année les degrés de température. Une observation sera une liste de 3 valeurs [ $\langle$ jour  $\rangle$ ,  $\langle$  mois  $\rangle$ ,  $\langle$  degré  $\rangle$ ]. Par exemple, une température de  $20^\circ C$  enregistrée le 13 février correspond à la liste [13, 2, 20]. Le but de l'exercice est de manipuler une liste de telles observations qui est une liste de listes et pour simplifier on supposera que toutes les observations sont enregistrées pendant des années non bissextiles (le nombre de jours du mois de février d'une année non bissextile est égal à 28).

### Travail demandé :

Soit la liste  $Nb\_Jour\_Mois=[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]$  contenant les nombres de jours des mois d'une année non bissextile. Cette liste, supposée déjà définie, sera passée comme paramètre en cas de besoin.

Répondre en MAPLE aux questions suivantes :

- 1) Ecrire la procédure **Valide** qui retourne *true* si à une observation **Obs** donnée correspond un mois et un jour corrects, et retourne *false* sinon.
- 2) Ecrire la procédure **Num\_Jour** qui permet de retourner le numéro du jour dans l'année d'une observation **Obs** donnée et supposée valide.  
*Exemple* : L'observation [12, 2, 10] enregistrée le 12 février correspond au 43<sup>ème</sup> jour de l'année.
- 3) Ecrire la procédure **Avant** qui, à partir de deux observations **Obs1** et **Obs2** supposées valides, retourne *true* si l'observation **Obs1** a eu lieu avant l'observation **Obs2** et retourne *false* sinon.
- 4) Ecrire la procédure **Ajout\_Obs** qui ajoute, à la fin d'une liste d'observations **L** donnée, une observation **Obs** donnée, si celle ci est valide et non déjà enregistrée.
- 5) Ecrire la procédure **Tri\_Obs** qui ordonne une liste d'observations **L** par ordre chronologique des observations.

# PROBLEME : Construction d'une table de symboles

Lors de l'analyse d'un programme source, les variables déclarées sont mémorisées dans une structure de données appelée **table de symboles**.

Dans ce problème, on propose de créer une table de symboles suite à la déclaration d'un nombre connu de variables. La table de symboles est représentée par les trois tableaux à une dimension suivants :

**Tnom** : tableau de caractères

**Ttype** : tableau d'entiers

**Tpos** : tableau d'entiers

Le tableau **Tnom** sert à mémoriser les noms de variables valides.

Un nom de variable valide est composé de 8 caractères alphanumériques au maximum (lettre majuscule ou minuscule, ou chiffre ('0'..'9')) et ne doit jamais commencer par un chiffre.

Un nom de variable est mémorisé en stockant les caractères le composant dans des cases consécutives de **Tnom** à raison d'un seul caractère par case. Les différents noms de variables sont séparés par le caractère '\*'. Le marqueur de fin du tableau **Tnom** est le caractère '#'. ...

Le tableau **Tpos** mémorise l'indice de début de chacune des variables dans **Tnom**. Le marqueur de fin du tableau **Tpos** est 0.

Le tableau **Ttype** mémorise les références des types des variables.

Dans la suite, uniquement les types suivants ainsi que leurs références seront utilisés :

|                    |         |        |      |           |
|--------------------|---------|--------|------|-----------|
| <b>Type :</b>      | Booléen | Entier | réel | caractère |
| <b>Référence :</b> | 1       | 16     | 32   | 8         |

Ainsi la figure 1 illustre le contenu de la table de symboles des variables **COUR** de type réel, **TOTAL** de type entier, **b1** de type booléen et **i** de type entier.

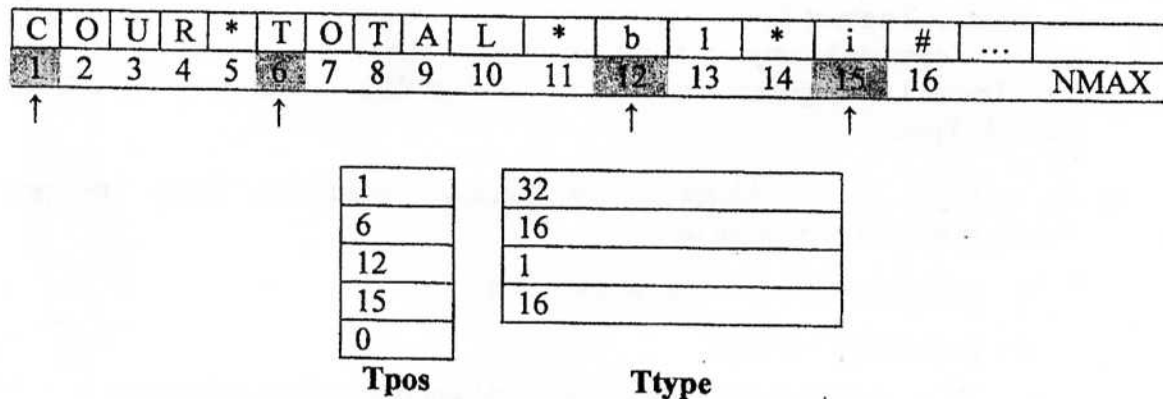


Figure 1. Mémorisation de la table des symboles

## Travail demandé :

Dans ce qui suit, on suppose avoir déjà effectué les déclarations suivantes :

**CONSTANTE**      NMAX=1000  
**TYPE**              TAB1 = tableau [1..10\*NMAX] de caractère  
                      TAB2 = tableau [1.. NMAX+1] de entier  
                      TAB3 = tableau [1..8] de caractère

En algorithmique une chaîne de caractères est manipulée comme un tableau de caractères. Uniquement les fonctions suivantes sont supposées prédéfinies :

- **Longueur(ch)** retourne la taille de la chaîne de caractères *ch*.
- **Car\_Alph(C)** retourne *vrai* si le caractère *C* est alphabétique et *faux* sinon.
- **Car\_Num(C)** retourne *vrai* si le caractère *C* est numérique et *faux* sinon.

Répondre en algorithmique aux questions suivantes :

- 1) Ecrire une procédure **Nb\_Elt** qui saisit et retourne un entier **nb** compris entre deux entiers **hinf** et **bsup** donnés.
- 2) Ecrire une fonction **Nom\_Valide**, à résultat booléen, qui retourne *vrai* si une chaîne de caractères **Ch** donnée représente un nom de variable valide et *faux* sinon.
- 3) Ecrire une procédure **Saisie\_Variable** qui saisit et retourne **V** de type chaîne de caractères représentant le nom d'une variable valide, ainsi que **T** la référence valide de son type.
- 4) Ecrire une procédure **Ajout\_Variable** qui ajoute une variable supposée valide dans la table de symboles suivie du caractère '\*'. Elle prend en entrée le nom de la variable **V** et **T** la référence de son type ainsi que **pos**, la position d'insertion dans **Tpos** et met à jour les tableaux **Tnom**, **Ttype**, **Tpos**.
- 5) Ecrire une procédure **Recup\_Nom** qui récupère dans un tableau de caractères **TC** de type **TABC** les caractères d'un nom de variable mémorisé, à partir d'un indice **pos**, dans le tableau **Tnom**.
- 6) Ecrire une fonction **Recherche**, à résultat booléen, qui à partir d'un nom de variable **V** et les tableaux **Tnom** et **Tpos**, retourne *vrai* si **V** existe déjà dans **Tnom** et *faux* sinon. La recherche consiste à comparer **V** avec les noms de variables de même longueur mémorisés dans **Tnom**. Les longueurs, des noms de variables déjà mémorisés, seront déterminées à partir de **Tpos**.
- 7) En utilisant les procédures et les fonctions adéquates, écrire une procédure **Table\_Symboles** permettant de :
  - saisir le nombre de variables à déclarer ;
  - pour chaque variable :
    - saisir son nom et la référence de son type sachant qu'un nom de variable ne peut apparaître qu'une seule fois dans **Tnom** ;
    - l'ajouter à la table des symboles (**Tnom**, **Tpos**, **Ttype**) en faisant les mises à jour nécessaires.